Integer programming for optimal yaw control of wind farms

Felix Bestehorn[1,2], Florian Bürgel[1], Christian Kirches[1], Sebastian Stiller[1], and Andreas M. Tillmann[1,3]

[1]TU Braunschweig, Institute for Mathematical Optimization, Universitätsplatz 2, 38106 Braunschweig, Germany
[2]now with Carl Zeiss SMT GmbH
[3]now at TU Clausthal, Institute of Mathematics, Erzstraße 1, 38678 Clausthal-Zellerfeld, Germany

Correspondence: Florian Bürgel (f.buergel@tu-braunschweig.de)

Abstract. It is well-known that wakes caused by the wind turbines within a wind farm negatively impact the power generation and mechanical load of downstream turbines. This is already partially considered in the farm layout. Nevertheless, the strong interactions between individual turbines provide further opportunities to mitigate adverse effects during operation, e.g., by repeatedly adjusting axial induction or yaw offsets to wind conditions. We propose a mathematical approach, that covers the farm by patterns based on a smaller, precomputable so-called upstream section, in the form of integer programming for faster globally optimal yaw control (under some mild assumptions like discretized yaw offsets, chosen size of upstream section, and homogeneous layout structure). While we prove the wind farm yaw problem to be strongly $\mathcal{NP}$-hard in general, we demonstrate through numerical experiments that our method enables optimal yaw control under real-world control update periods. In particular, the approach remains efficient if turbines are deactivated and scales well with increasing farm width.

Keywords. Wind farm, optimization, yaw-based control, integer programming.

1 Introduction

Wind turbines are considered one of the most important electric power plants of the future energy grid, since they can generate electricity cheaply and with low greenhouse gas emissions, see, e.g., Kost et al. (2018). Naturally, they are placed in wind farms at locations with desirable year-round wind conditions (on- or offshore), to ensure a high energy yield in general. To increase efficiency during operation, turbines are *controlled*, i.e., parts of it are periodically adjusted (e.g., nacelle direction relative to the wind, i.e., the so-called yaw-based control, generator torque or blade pitch angles) to the most beneficial positions with respect to the wind speed and direction. The conventional control consists of a greedy strategy in which a turbine maximizes its own power output under certain durability considerations, see, e.g., Hau (2013). In a farm, such greedy control can lead to suboptimal total power output (as well as to increased maintenance effort for the turbines) as the turbines influence one another due to their spatial proximity: each turbine causes a wake, which is characterized by decreased wind speed and increased turbulence, and impacts downstream turbines regarding both power output and mechanical load (wear and tear). The control of a turbine affects the length and the spatial distribution of its wake. This opens the possibility for a *global* control that incorporates turbine interactions within the entire farm. In general, such a farm flow control is state of the art, see, e.g., Meyers et al. (2022). We focus on the optimization of the yaw offsets, which deflect and deform wakes, see, e.g., Annoni et al. (2018), as in Fig. 1, to maximize the farm's total power, which is a primary objective, see, e.g., van Wingerden et al. (2020).

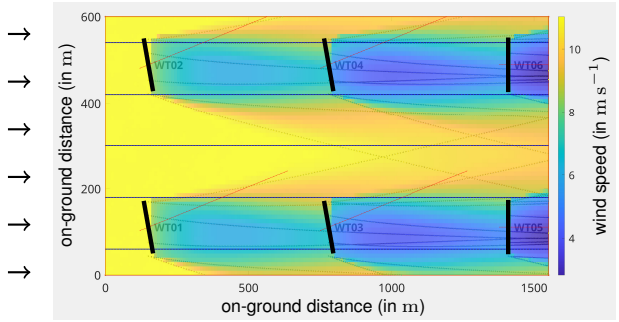


Figure 1. Simulation of the local wind speed (in $\mathrm{m\,s^{-1}}$) with 6 turbines (of type NREL 5-MW) arranged in a 2×3 grid layout; axes give on-ground distances (in m). The wind blows from west to east with a speed of $11\,\mathrm{m\,s^{-1}}$ with a turbulence intensity of 6%. It is decreased by the turbines and, additionally, the wakes are deflected by the turbines in the first and second column because of their yaw offset of $15°$. (The downstream-most turbines have a yaw offset of $0°$.) This figure was produced using the software WinFaST, see Sect. 3.1 for a description.

In the following, we discuss the wake models, farm layouts and operations to motivate the wind farm yaw problem and paraphrase our approach along with contributions and limitations; an outline of the remaining paper concludes this introduction.

The remaining paper is structured as follows: in Sect. 2, we formulate the *wind farm yaw problem* (WFYP) mathematically, we motivate and develop our *covering approach* (CA) and the corresponding *integer program* (IP). The details on precomputations, i.e., on the simulation and the resulting performance indicators are described in Sect. 3. We explain and discuss the results of our computational experiments in Sect. 4 and finally, conclude in Sect. 5. The theoretical complexity of the WFYP is examined in Appendix A. We abbreviate farm for wind farm, turbine or WT for wind turbine, and TI for turbulence intensity.

1.1 Related work

Wake models in the literature are often parameterized to NREL 5-MW turbines, see Jonkman et al. (2009) for details, in usual atmospheric conditions for onshore wind farms, see Sect. 3.1 for details. The following brief survey refers to this turbine type; for a detailed overview of the complex topic of wind farm flow control, we refer to, e.g., Meyers et al. (2022). High-fidelity simulations of wind farms in 3D like the *Simulator fOr Wind Farm Application* (SOWFA), see Churchfield et al. (2012a, b); Fleming et al. (2013), are time-consuming and, hence, impractical for use in control. An overview of the most important control-oriented models in 2D is given in (Annoni et al., 2018, Sect. 2.1): first, the Jensen model, see Jensen (1983); Katic et al. (1987) (also for a further developed model); second, the multi-zone model FLORIS, see Gebraad et al. (2014), which has the dynamic extension FLORIDyn, see Gebraad and van Wingerden (2014); and third, the Gaussian model, see, e.g., Bastankhah and Porté-Agel (2016). These models are continuously being developed further. In the present work, we use a simulation software which utilizes a wake model based on FLORIDyn, cf. Sect. 3. State-of-the-art static wake models are the Gauss-Curl-Hybrid model, which incorporates secondary effects of wake steering, see King et al. (2021), and the cumulative-curl wake model if there are more than a few wake interactions, see Bay et al. (2023). Both are implemented in the rapidly evolving software FLORIS (not to be confused with the original model), incorporating several steady-state control-oriented

wake models, see NREL (2024). These control-oriented wake models have limitations: for a comparison between FLORIDyn and SOWFA we refer to Gebraad and van Wingerden (2014). In any case, the mathematical approach we will propose utilizes a superordinate model in which the underlying wake model is interchangeable.

A profitable wind farm needs a suitable location and careful planning of turbine numbers and placement. *Wind farm layout optimization* relies on yearly wind frequency data (wind direction and speed). It has been known for decades that it is useful to avoid wake-induced yield reductions of downstream turbines by setting them far enough apart, see, e.g., Katic et al. (1987). While first attempts to account for such effects merely simulated the annual average output of a specific farm layout, see Katic et al. (1987), in recent years, the layout problem was optimized globally by mixed-integer and constraint programming, see, e.g., Zhang et al. (2014); Fischetti (2017, 2021). In addition to the annual energy output, noise propagation is a concern in case of onshore farms that can also be considered, see Zhang et al. (2014). For offshore farms, aspects of cable routing and jacket foundations are worth taking into account, see Fischetti (2017); Fischetti and Pisinger (2019).

For a given wind farm (i.e., a fixed layout), it is conventional to run greedy control for each individual turbine, see, e.g., Hau (2013). As mentioned earlier, adopting a global control for the whole farm instead of local control of separate turbines can potentially improve the overall power output and relieve physical strain on the turbines. In general, there are two common global control concepts, cf., e.g., Gebraad et al. (2015); Annoni et al. (2016); Meyers et al. (2022): axial induction-based control (of generator torque and/or the collective blade pitch angle) and yaw-based control (of the turbine yaw offsets), which is also known as wake steering control, see, e.g., Howland et al. (2019). Both control concepts effectively reduce power generation of upstream turbines by adjusting torque/pitch or yaw, respectively, which in turn leads to increased wind speeds (relative to those under greedy control) in their respective wakes and, consequently, higher power yield of the affected downstream turbines. The main aim of these concepts is to achieve a net gain, and even small improvements are deemed promising, see, e.g., the wake steering study by Howland et al. (2022) with average power increases of 0.3 to 2.7% for a commercial wind farm.

In general, it depends on the allocation of the turbines, their characteristics, and the wind conditions whether a control different from the greedy one can indeed yield the desired gains. For example, it may happen that wind speeds are so high that all turbines operate at maximal capacity anyway; nevertheless, some control could then still be meaningful to reduce mechanical loads. Furthermore, there are cases in which axial induction control shows no positive effect on total power output while yaw control yields significant improvements, see the high-fidelity computational fluid dynamics simulations in Gebraad et al. (2015). Thus, we will focus on yaw control in this paper, where changing the yaw offset of a turbine deflects its wake.

There is already a lot of research on yaw-based control. We follow the literature distribution in Stanley et al. (2022), which divides it into two parts to tackle the optimization problem, i.e., using continuous yaw offsets between lower and upper bounds, see Gebraad et al. (2014); Fleming et al. (2016); Gebraad et al. (2017), and using discretized yaw offsets, see Dar et al. (2017); Dou et al. (2020). In Gebraad et al. (2014) a slow game-theoretic approach is used, which does not necessarily deliver a global optimum as desired. This is also not delivered in Fleming et al. (2016); Gebraad et al. (2017) as their optimization method is based on sequential quadratic programming (SQP). However, a combination of yaw control and farm layout optimization has been considered in both references, which is an interesting application but out of scope of the present paper. In Dar et al.

(2017), the authors modified the Jensen model (cf. Jensen (1983)) to include the effect of yaw offset adjustments and developed a dynamic programming formulation (DPF) to pass the wind speed downstream from turbine to turbine, which results in a very efficient method for turbines in a single row. However, the nonlinearity of the equation to compute the wind speed for a turbine located in several turbine wakes prevents transferring this concept to a 2D farm layout. Nevertheless, Dar et al. (2017) showed that optimizing each row separately with DPF, i.e., ignoring effects in adjacent rows, reduces the gap between its so-called wind farm efficiency and its globally optimal quantity (obtained by full enumeration) to up to 1%. This idea to split the farm into disjoint subsets of turbines has already been presented in Spudić and Baotić (2013), which tackles distributed systems, and is also used in Siniscalchi-Minna et al. (2020); Bernardoni et al. (2021); Dong and Zhao (2023). The latter one even allows at least one turbine per subset to be in several subsets, which can lead to conflicting yaw offsets: while they describe a helping logic to handle this, we inherently ensure the yaw offset compatibility for any number of turbines. Our goal is to determine the global optimum—under some mild assumptions, see Sect. 1.2.1 for details—without resorting to full enumeration; for this, our superordinate model takes dependencies of adjacent subsets of turbines into account and can integrate any kind of wake effect simulation. In fact, we will consider the farm as a network. This general point of view has already been used in Annoni et al. (2019) for a different application, namely to share information among nearby turbines to improve wind direction estimation; the underlying method, which allows simultaneous clustering and optimization on graphs, was developed in Hallac et al. (2015). In Dou et al. (2020), the covariance matrix adaptation evolution strategy is employed, which is a heuristic algorithm for black-box functions. In contrast, our focus is to exploit the structure of the optimization problem. In Stanley et al. (2022), the structure of the problem is used in the new so-called Boolean approach. This considers which turbines have downstream ones in their wake, starts at the upstream-most turbine, and fixes a yaw offset to either 0° or 20° if it increases the power of the farm; the simulations used the software FLORIS (cf. NREL (2024)) with the Gauss-Curl-Hybrid wake model (cf. King et al. (2021)). In the adaption—called serial-refine method—, see Fleming et al. (2022), each turbine is run through twice (in a serial and a refine pass), which allows several yaw offsets. This method is fast as well as successful and suitable for a comparison even if it generally does not guarantee a globally optimal solution. In contrast, our approach optimizes yaw offset settings simultaneously for the whole farm. (From the view of blade load, which we do not take into account, a slightly positive yaw offset is best, whereas the exact location depends on level of wind shear, see the field study Ennis et al. (2018).)

1.2 Contributions and limitations

We provide a method of globally optimal yaw control (under some mild assumptions) that also includes the possibility to deactivate wind turbines, e.g., for maintenance reasons. To that end, we propose a novel superordinate model which exploits the coupled nature of wind turbines in a wind farm and can be based on any wake model.

We refer to the determination of a globally optimal combination of yaw offsets for a given wind farm layout and given wind conditions (i.e., subject to arbitrary but fixed wind speed and direction) as a WFYP; see Sect. 2 for details and a mathematical problem definition. In this context, global optimality refers to an objective function, which takes the total power output of the farm into account—our main goal—and can include other quantities representing mechanical loads; in fact, we include the important tower load and the pitch angle changes (causing some wear) as so-called tower activity and pitch activity, see Sect. 3

for a definition; we do not consider the blade load as the used simulation software, described in Sect. 3.1, does not provide a suitable output quantity.

The complex nonlinearities of wake flow dynamics and turbulence are typically only available through simulation, which makes a direct integration into an optimization model almost impossible. A naive approach to solve the WFYP is to run simulations for *every possible* yaw offset combination, i.e., full enumeration, but is already impractical for small farms. The crucial observation is that the wake interactions of turbines adhere to certain patterns with respect to the farm layout that occur repeatedly, and with overlaps, across the entire farm. We exploit these redundancies to greatly reduce the number of required combinations: our superordinate model constructs the farm on the basis of these patterns of depending turbines and ensures the consistency of selected yaw offsets in regions of overlapping patterns; we formulate a corresponding IP to receive the desired yaw offsets as solution.

Our numerical experiments are intended as *proof of concept* as we use error-free simulation data. They will show that state-of-the-art solver software for this problem class—e.g., Gurobi, see Gurobi Optimization, LLC (2022), or SCIP, see Bestuzheva et al. (2021)—can solve these WFYP problems within reasonable time, demonstrating the practicality of our approach.

The use of our superordinate model enables deactivating any turbines and a large scaling of the wind farm size orthogonal to the wind direction, whereas a scaling in wind direction significantly increases the computational effort due to a stronger growth of relevant patterns—while beyond the scope of this paper, scaling in wind direction is possible by the following idea: split the patterns in segments in wind direction (rows of turbines so to speak), compute the upstream-most segment with specific yaw offsets, run the simulation, save the resulting wind field, use it to simulate all yaw offset combinations in the next segment and so on. Moreover, the weighting flexibility of the objective function terms (power output and mechanical loads) allows additional objectives to be considered, e.g., putting selected turbines in a low-load operating mode.

1.2.1 Assumptions

We consider the setting in which all turbines in the wind farm are of the same type and arranged on an underlying irregular grid. These assumptions are not particularly restrictive in practice since, on the one hand, turbines within one farm are typically of the same type—although layout optimization can result in different turbine heights, see, e.g., Stanley et al. (2017)—and on the other hand, we can, in principle, choose the grid resolution as fine as needed to allow representing any layout (by leaving some grid points unused); e.g., the results of layout optimization in Thomas et al. (2015); Gebraad et al. (2017) are not arranged on a simple grid. However, an irregular grid with, e.g., three and five rotor diameters distance between the turbines is quite common, see, e.g., Gebraad et al. (2014); Gebraad and van Wingerden (2014); Boersma et al. (2018, 2019b).

Moreover, our model of the WFYP problem relies on two operational assumptions, which can also be realized with arbitrary fine resolution and should therefore not be restrictive in applications: the admissible yaw offsets are bounded—to prevent overly strong mechanical loads, cf., e.g., Boersma et al. (2019a)—as well as discretized, and we impose a threshold below which the influence of wakes on downstream turbines is deemed negligible.

In particular, Fleming et al. (2016) limit the yaw offset to $[-45°, +45°]$ and $[-25°, +25°]$, Boersma et al. (2019a) to $[-25°, +25°]$, Stanley et al. (2022) to $[0°, 30°]$, and our industry partner suggests $[-15°, +15°]$ to protect the turbines. In

our computational experiments, e.g., we choose yaw offsets from $[-15°, +15°]$ at $5°$ increments and set the downstream-most turbines to $0°$ (to reduce the number of options, see Sect. 2.1)—we always state yaw offsets relative to the (fixed) wind direction, i.e., as yaw offsets with respect to the mathematically positive sense of rotation—and disregard wake influence if the wake-induced wind speed reduction (relative to the given speed) at the downstream turbine is 5% or less. We will also refer to this exemplary setup for illustrative purposes when we formally define the WFYP and our solution approach in Sect. 2. Nevertheless, our method admits arbitrary other settings, e.g., discretized yaw offsets for the downstream-most turbines.

Our choice is not unrealistic: Quick et al. (2020) describe the problem of uncertainty of incident wind conditions for metrological reasons and for real-world causes; in fact, the inflow of a wind farm can consist of several wind directions, speeds, TIs and shears (e.g., caused by a mountain). Stanley et al. (2022) deduce that it is unrealistic to solve the WFYP with continuous or finely discretized yaw offsets and choose their Boolean optimization approach only deciding whether a turbine is yawed or not, i.e., set $0°$ or $20°$ (which is a result of power simulations of turbines with a yaw offset discretization of $5°$); they also compared their approach with a common continuous yaw optimization (based on gradients) and mostly achieve the same power improvement. Against this background, our choice of $5°$ increments is reasonable.

In general, wind turbulence depends on a number of meteorological and topographical factors, whereas the power output essentially depends on long-term fluctuations and loads are caused by short-term fluctuations, see Hau (2013). We will fix the TI, see Sect. 3.1 for a definition, throughout this paper; for an impression of the locally strong speed fluctuations in the wind field, see Fig. 1. For general effects we refer to Talavera and Shu (2017): first, there is a correlation between the increase of TI and faster wake recovery (as wind speed recovers faster for turbulent shear flow in comparison to laminar shear flow) and second, turbulent inflow increases the power output of a wind turbine (because of suppressed flow separation).

The correlation between important weather characteristics like temperature, relative humidity, wind speed and wind gusts are investigated in (Vladislavleva et al., 2013, Fig. 2): as expected, wind speed and gusts have a strong positive correlation with the power output while the pressure has a slightly negative one. Finally, the spectrum of possible influences is wide. We focus here on the most influential factors, i.e., wind speed and direction, and fix the others like TI and air density for simplicity.

1.2.2 Complexity theory point of view and computation time

In fact, while the described homogeneous turbine type as well as layout structure, and the yaw offset discretization, may seem to simplify the problem, this is not the case from the viewpoint of computational complexity theory: as we will prove in Appendix A, the WFYP is generally $\mathcal{NP}$-hard, which means that an efficient solution algorithm—i.e., one with run time polynomially bounded by the input size—is highly unlikely to exist, cf. Garey and Johnson (1979). This computational intractability result, along with discretization-related aspects, motivates and justifies tackling the WFYP by IP techniques; see, e.g., Schrijver (1986) for a thorough introduction to IPs.

Thus, the only remaining potentially limiting aspect is the computation time, which naturally increases with growing problem size and complexity. The assessment of Fleming et al. (2022) for a real-time control scale is of seconds to minutes. Nevertheless, in general, we can exploit two mitigating facts: first, it is useful to avoid continuous small yaw movements in order to not unduly increase mechanical loads, and second, the yawing rate must be slow (approximately $0.5\,° \,\mathrm{s}^{-1}$) to avoid gyroscopic moments,

see (Hau, 2013, Sects. 6.3.1 and 11.3). As a consequence, we assume that a computation time of less than $1\,\mathrm{min}$ is sufficient; in addition, a light detection and ranging (LIDAR) system could provide wind information at an early stage. As our IP approach is capable of determining globally optimal yaw offset combinations for entire farms of considerable size at most under one minute (after required precomputations), e.g., 27 wind turbines in $11\,\mathrm{s}$, it is suitable for real-world application. However, the practical realization of a short real-time control scale would require a time-delayed yaw offset adaption of each turbine (depending on the propagation time). In our computational experiments, we will not consider this transient phase in the but the effects over a longer period of time instead, namely $10\,\mathrm{min}$, see Sect. 3.2 for details.

2 The wind farm yaw problem

Recall that our WFYP aims to find a set of yaw offsets that maximizes the total power output of the farm, optionally along with other quantities, under a given wind scenario, i.e., fixed wind direction and speed. We will formalize this optimization task in Sect. 2.1, develop our CA in Sect. 2.2, and derive the corresponding IP in Sect. 2.3.

2.1 Notation, basic WFYP formulation as IP with black-box objective, and the curse of dimensionality

We consider a farm with n_{WT} turbines, each identified by an index from the set $T := \{1, 2, \dots, n_{\mathrm{WT}}\}$. Later, in Example 2.2, we will use our ultimate assumption that turbines are located on an irregular grid with certain spacing in both dimensions. We assign to each turbine $i \in T$ a set Γ_i of admissible yaw offsets, see Sect. 1.2.1, with respective size $n_{\Gamma,i} := |\Gamma_i| < \infty$. For every turbine i, we associate an index set $N_{\Gamma,i} := \{1, \dots, n_{\Gamma,i}\}$ with its admissible yaw offsets. This general notation allows for turbines of different types, but even when working with identical ones, for which the yaw offset sets usually coincide, difference may arise, e.g., if maintenance reasons limit the options for specific turbines.

Recalling that the turbines can influence each other, the overall power output of the farm and load-related other quantities depend on the global yaw configuration, i.e., the collection of the set yaw offsets of all individual turbines, as well as the considered wind conditions (in particular, direction and speed). Since the precise relation of these aspects has no known analytical form, the objective function of the WFYP must generally be considered a *black-box* whose values for a specific combination of input parameters can be determined, or estimated, by running a simulation of the corresponding farm scenario.

To specify a basic mathematical formulation of the WFYP, we introduce binary decision variables $x_{i,j}$ for all $i \in T$, $j \in N_{\Gamma,i}$; if turbine i is set to the yaw offset (from Γ_i) indexed by j, then $x_{i,j} = 1$, and otherwise $x_{i,j} = 0$. As any turbine can only operate with one yaw offset at a time, these decision variables must adhere to $\sum_{j \in \Gamma_i} x_{i,j} = 1$ for all $i \in T$. The black-box objective can then be described by a function $f_\omega \colon \{0,1\}^{n_{\Gamma,1} + \dots + n_{\Gamma,n_{\mathrm{WT}}}} \to \mathbb{R}^{n_{\mathrm{WT}}}$, where we omit the dependency on the (here, fixed) wind direction and speed as well as farm layout for notational convenience and where the vector ω consists of two weighting factors, which we will discuss later. This function is comprised of the objective contribution of every turbine, which is impacted by its own yaw offset as well as the yaw configuration of the remaining farm (in fact, not all other turbines influence any given one, but for now, we do not utilize this). In particular, the function $f_\omega(x) = (f_{\omega,1,j_1}(x), \dots, f_{\omega,n_{\mathrm{WT}},j_{n_{\mathrm{WT}}}}(x))^\top$, where $f_{\omega,i,j}(x)$ denotes the objective contribution of turbine i when set to yaw offset index j from its admissible set Γ_i (as per $x_{i,j} = 1$). Here,

the yaw configuration of all turbines (in particular those that influence i) is fixed as prescribed by the decision variables x. To find the optimal yaw configuration of all turbines, i.e., a solution $x \in \{0,1\}^{n_{\Gamma,1}+\cdots+n_{\Gamma,n_{\mathrm{WT}}}}$, our objective function sums up contributions of individual turbines:

$$f_\omega^\Sigma(x) := \sum_{i=1}^{n_{\mathrm{WT}}} \sum_{j=1}^{n_{\Gamma,i}} f_{\omega,i,j}(x)\, x_{i,j} \quad \text{with black-box simulation function} \quad f_{\omega,i,j}(x) := P_{i,j}(x) - \omega^{(\mathrm{T})}\, a_{i,j}^{(\mathrm{T})}(x) - \omega^{(\mathrm{P})}\, a_{i,j}^{(\mathrm{P})}(x), \quad (1)$$

where $P_{i,j}$, $a_{i,j}^{(\mathrm{T})}$ and $a_{i,j}^{(\mathrm{P})}$ are the average power output, the tower activity, and the pitch activity—all over a certain (fixed) period of time—of turbine $i \in T$ with yaw offset index $j \in \Gamma_i$ (and the remaining yaw offsets corresponding to the selections encoded in x). To evaluate the black-box for a specific x, one needs to resort to *simulation* to obtain the power and mechanical load values for each turbine in the considered farm (under the given wind scenario)—the details, including definitions of the notions of tower and pitch activity, will be described in Sect. 3. The (nonnegative) weighting factors $\omega^{(\mathrm{T})}$ and $\omega^{(\mathrm{P})}$ are set a priori and determine the relative importance of the respective quantities in the optimization objective; in particular, both weights can be set to zero to take only the power into account; in addition, we could choose individual weights for each turbine.

Thus, we formulate the WFYP as an IP with black-box objective:

$$\max_x \quad \sum_{i=1}^{n_{\mathrm{WT}}} \sum_{j=1}^{n_{\Gamma,i}} f_{\omega,i,j}(x)\, x_{i,j} \quad (2)$$

$$\text{s.t.} \quad \sum_{j=1}^{n_{\Gamma,i}} x_{i,j} = 1 \qquad \text{for } i = 1,\ldots,n_{\mathrm{WT}} \quad (3)$$

$$x_{i,j} \in \{0,1\} \qquad \text{for } i = 1,\ldots,n_{\mathrm{WT}} \quad \text{and} \quad j = 1,\ldots,n_{\Gamma,i}. \quad (4)$$

Due to the black-box nature of the objective function, the above formulation cannot simply be handled by off-the-shelf IP solvers. Indeed, we call it the "basic" formulation because it essentially requires computing *all* $f_{\omega,i,j}(x)$ to obtain a standard (non-black-box) IP, and hence corresponds to the naive brute-force full enumeration. Clearly, this approach is only viable for very small WFYP instances—i.e., few turbines with a small set of admissible yaw offsets—due to the exponential growth of yaw offset combinations; see also Remark 2.3 given below. Moreover, each simulation run incurs a certain run time that itself increases with the farm size. Thus, the WFYP suffers from the typical "curse of dimensionality" often encountered in combinatorial problems. In fact, our following result establishes that an efficient (polynomial-time) solution method for the WFYP very likely does not exist; the proof is deferred to Appendix A.

Proposition 2.1 (Theorem A.3 and Corollary A.4 from Appendix A)**.** *The WFYP is strongly $\mathcal{NP}$-hard and cannot be approximated within any factor $\alpha \leq 1$ in polynomial time (unless $\mathcal{P} = \mathcal{NP}$).*

Example 2.2 (A 3×2 farm)**.** We consider $n_{\mathrm{WT}} = 6$ turbines, arranged in a 3×2 farm, see Fig. 2; we assume the wind blows from west to east and we identify the turbines with the index set $T = \{1,\ldots,6\}$. The turbines may be homogeneous of type NREL 5-MW turbines with a rating value of $5\,\mathrm{MW}$ and a rotor diameter of $D = 126\,\mathrm{m}$, cf. (Jonkman et al., 2009, Table 1-1). We set the turbine spacing to $3D \times 5D$, i.e., turbines are on an irregular grid with three and five rotor diameters distance

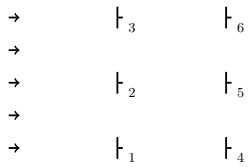


Figure 2. Illustration of the 3×2 farm layout from Example 2.2; the wind direction is indicated by small arrows. The drawing is true to scale for NREL 5-MW turbines, which have a rotor diameter of $D = 126\,\mathrm{m}$, and for a turbine spacing of $3D \times 5D$.

between turbines in the width ("column") and depth ("row") direction, respectively. We choose the notation analogous to matrices; in the literature, our example would usually be referred to as 2×3 with $5D \times 3D$. However, the spacing choice is common, see, e.g., farms in Gebraad et al. (2014); Gebraad and van Wingerden (2014); Boersma et al. (2018, 2019b) (also, Katic et al. (1987) mentions $5D$ as row value but no column value). We restrict the permissible yaw offsets to $\gamma_i \in \Gamma_i = \Gamma := \{-15°, -10°, \ldots, +15°\}$ and $n_{\Gamma,i} = n_\Gamma := 7$ for every $i \in T$ (cf. Sect. 1.2.1).

Remark 2.3 (Total number of possible yaw configurations of a 3×2 farm)**.** *Example 2.2 yields a total number of* ${n_\Gamma}^{n_{\mathrm{WT}}} = 7^6 = 117\,649$ *possible yaw configurations. Consequently, this number of farm simulations would be required to solve the WFYP with the basic approach for one wind scenario. Therefore, we need a different approach to reduce the number of simulations. A coarser yaw offset discretization is not an option as it would sacrifice the level of exercisable control. Indeed, our approach achieves this by reducing the number of yaw offset combinations to consider and by reusing simulation results where possible.*

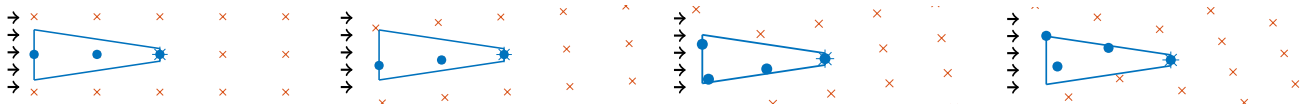
A turbine has the highest power output with a yaw offset of $0°$ (i.e., it runs greedily), see, e.g., Dar et al. (2017). So, an initial approach that most likely preserves the most power output but reduces the number of options is to let the downstream-most ones run greedily. (We verified by an experiment that the wake of a yawed turbine has no influence on this approach at distance of $3D$.) The number of possible yaw configurations in Example 2.2 would then reduce to $7^3 = 343$; however, such an approach does not scale—a 3×3 farm again results in 7^6 configurations. This emphasizes the need for an altogether different approach.

2.2 Covering approach for WFYP solution

The assumed homogeneous turbine type and layout structure gives rise to recurring patterns, that can be exploited to equivalently reformulate the WFYP in a way that reduces the number of black-box evaluations (i.e., simulation runs).

2.2.1 Upstream sections

To that end, we take a closer look at the turbines that *influence* a specific turbine, i.e., affect the latter by the downstream wind wake; we call the set of turbines that *influence* a specific one as an *upstream section* (including the specific one). The size depends on the wind conditions (in particular, the wind direction) and the yaw offset(s) of the influencing turbine(s). It includes all turbines that *could* influence as upstream section, the specific turbine in focus, i.e., the upstream section is based on the admissible yaw offset *ranges*. Moreover, we remind the reader that one assumption is to disregard wake influence if the wake-



(a) Upstream section for $0°$. (b) Upstream section for $5°$. (c) Upstream section for $10°$. (d) Upstream section for $20°$.

Figure 3. Upstream sections (blue trapezoids) with included turbines (marked as points; asterisks are downstream-most turbine in upstream section) in a grid of turbines (red crosses). The grid represents a farm with infinite expansion.

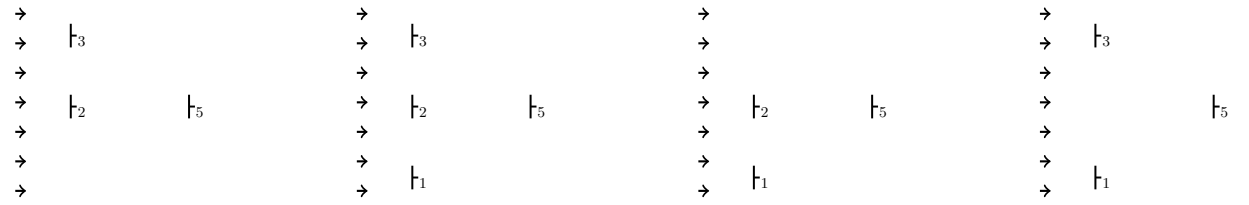
induced wind speed reduction (relative to the free stream) at the downstream turbine is at a threshold or below—throughout this paper, 5%, see Sect. 1.2.1. The concrete chosen area is a trapezoid (for visualization we sometimes use triangles) based on this threshold (without guaranteeing it): for this, we simply use the wind speeds at the so-called observation points (from simulations with $0°$ and extreme yaw offsets, e.g., $\pm 15°$) in the wake, see Sect. 3.1, and set the absolute value of the slope of the trapezoid to at least 0.15, which corresponds to an angle of approximately $8.5°$, see Fig. 3. (For a rough comparison, Dar et al. (2017) neglect the deficit in velocity for yaw offsets beyond $20°$ from the center of the turbine based on Jensen (1983).) Finally, the depth of the trapezoid constructed as described above is usually truncated (or, rarely, extended) to match the depth of the farm—in fact, we use the *effective depth of the farm*, which we define in Sect. 2.2.3; it coincides with the depth of the farm if the wind direction is $0°$ and can be smaller otherwise.

To illustrate upstream sections, reconsider Example 2.2 and its farm in Fig. 2. Under these fixed wind conditions, WTs 1, 2, and 3 influence WT 5 depending on the yaw offsets in $[-15°, +15°]$, i.e., the upstream section at WT 5 is given by $\{1, 2, 3, 5\}$. The chosen upstream section is useful to explain the concept; in fact, with the selected experimental setup for our results in Sect. 4 the corresponding upstream section would only include WTs 2 and 5, see Fig. 3(a) for a corresponding farm with three turbines in depth; and Fig. 3 for an overview with wind directions of $0°$, $5°$, $10°$ and $20°$.

2.2.2 Section configurations

Depending on the positions of subsets of turbines within the farm (keeping all other aspects fixed), upstream sections can take on different patterns, which can be identified based solely on the grid layout of the farm, see, e.g., Fig. 4. In particular, we can omit (or deactivate) turbines within any upstream section, thereby obtaining what we call *section configurations* as structural subsets of the complete section configuration, i.e., the upstream section itself. Crucially, if all turbines are of the same type, only a single upstream section is needed as a "template" from which to extract the appropriate "patterns" with which the farm can be represented—i.e., we can *cover* the entire farm using (overlapping) section configurations—and simulations can focus on the area of upstream section (reusing simulation results of the section configurations) rather than the whole farm directly. After the following example, we will formalize and explain this CA.

We can cover the 3×2 farm from Example 2.2 (cf. Fig. 2) by those section configurations shown in Fig. 4: we anchor the configuration from (b) at WT 5 and the ones from (a) and (c) at WTs 4 and 6 (as the respective downstream-most turbine instead of 5), respectively, since the corresponding parts of the farm exhibit the same structural pattern, see also Fig. 5(a). A change of the farm layout would require other section configurations; e.g., without WT 2 we would need the configuration



(a) Section configuration with inactive WT 1.

(b) Complete section configuration (i.e., upstream section).

(c) Section configuration with inactive WT 3.

(d) Section configuration with inactive WT 2.

Figure 4. Example of some section configurations of the upstream section with $n_{\mathrm{WT},u} = 4$ turbines. We kept the numbering from Fig. 2, but the depicted patterns may and do occur in other parts of the farm as well.

depicted in Fig. 4(d) for WT 5, and the one with just two turbines directly behind each other (not depicted) for WTs 4 and 6. In total, for the upstream section, we have 16 possible section configurations, including the complete one and the empty one. In general, if an upstream section encompasses $n_{\mathrm{WT},u}$ turbines, there is a total of $2^{n_{\mathrm{WT},u}}$ possible section configurations.

As in the example, we only need a small number of the possible section configurations to cover the farm during normal operation. However, we take into account all possible section configurations: it increases the precomputation time but preserves flexibility, i.e., we are prepared for deactivated turbines and can enlarge the farm orthogonal to the wind direction.

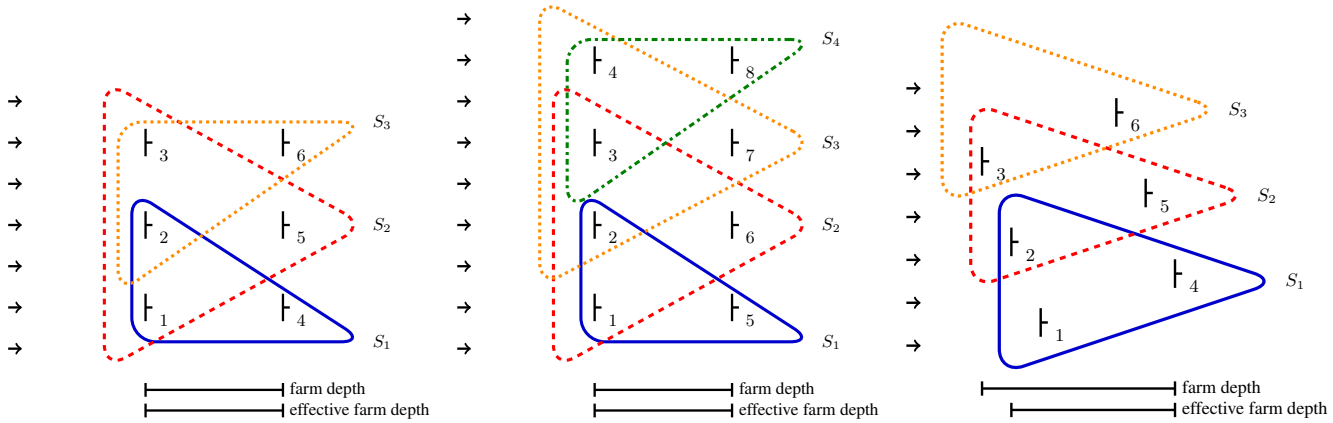
2.2.3 Covering sections to reduce the computational burden

To formalize the notion of section configurations that are suitable for covering the farm, it suffices to focus on the downstream-most turbines, whose number we denote by n_{s}, and determine so-called covering sections anchored at them.

Definition 2.4 (Covering sections)**.** A *covering section* is a set $S_k \subseteq T$ of turbines in a farm that influence each other with respect to (wake) disturbances. We denote the *set of covering sections* in a farm by $S := \{S_1, \ldots, S_{n_{\mathrm{s}}}\}$. Furthermore, we denote the set of those covering sections that contain a specific turbine i by $S(i) := \{S_k \in S : i \in S_k\}$.

To cover the farm, we must assign one covering section to each downstream-most turbine, as illustrated in Fig. 5, but as different covering sections can have the same pattern, a section configuration can be used several times, e.g., in Fig. 5(b). The core advantage of this CA is the significantly reduced number of simulations required to find the best WFYP solution (in comparison to full enumeration) as we only need to precompute the yaw configurations within every (distinct) section configuration and, accordingly, obtain the simulation results for all covering sections.

We already mentioned that wind directions deviating from $0°$ require to define the *effective depth* of the farm. For illustration, we use a 3×2 farm with a wind direction of $0°$ and $20°$, see Figs. 5(a)/(c). In the case of $0°$, the farm and the corresponding upstream section both have a depth of $5D$. For other directions, e.g. as in Fig. 5(c), this depth changes. The *depth of the farm* is the distance in x-direction between the upstream- and downstream-most turbines *inside the farm*, i.e., in our example between WTs 3 and 4. The *depth of the upstream section* is analogously defined *inside the upstream section*—in our example, WTs 2 and 4. As this depth is sufficient to finally cover the farm we also call it *effective depth of the farm*. Usually, our use of terms "upstream-" and "downstream-most" turbines refers to these covering sections, e.g., in our example, WTs 4 to 6



(a) In a 3×2 farm: S_1 (anchored at WT 4) uses the SC from Fig. 4(a) to cover WTs $\{1,2,4\}$, S_2 (at WT 5) employs the SC from Fig. 4(b) for $\{1,2,3,5\}$, and S_3 (at WT 6) uses the SC in Fig. 4(c) for $\{2,3,6\}$.

(b) In a 4×2 farm: S_1 (anchored at WT 5) uses the SC from Fig. 4(a) to cover WTs $\{1,2,5\}$, both S_2 and S_3 (at WTs 6 and 7, resp.) use the SC from Fig. 4(b) for $\{1,2,3,6\}$ and $\{2,3,4,7\}$, resp., and S_4 (at WT 8) for $\{3,4,8\}$ by the SC from Fig. 4(c).

(c) In a 3×2 farm with a wind direction of 20°: S_1 to S_3 are sufficient to cover the farm although their depth coincides with that of the upstream section, which is smaller than that of the farm.

Figure 5. Covering sections $S_1, \ldots, S_3$ in a 3×2 farm and $S_1, \ldots, S_4$ in a 4×2 farm, respectively, where S_1 is outlined by the solid blue, S_2 by the dashed red, S_3 by the dotted orange, and S_4 by the dash-dotted green line. Subfigure captions specify which section configuration (SC) is used by the covering section at a turbine and which turbines are covered by that.

are the downstream-most ones and serve as anchors for the covering sections. If an anchor turbine is missing, say WT 4, we relocate the covering section, i.e., in our example, we attach S_1 at WT 1 (where S_1 uses the section configuration with only one active turbine). Then we can assume without loss of generality that the downstream-most turbine inside a section configuration is always active, thus circumventing the half of all combinations in which the downstream-most turbine could be inactive. (Alternatively to this relocation, one could have anchored the covering section at a deactivated "virtual" WT 4.)

2.2.4 The required number of simulations

Before we turn to the WFYP model based on the CA, we take a closer look at the required number of simulations to solve it. For simplicity, we assume the same set of admissible yaw offsets, say, Γ with $n_\Gamma := |\Gamma|$ for the (identical) turbines. In the basic approach for the whole farm with n_{WT} turbines, we saw below Remark 2.3 that the total number of distinct yaw configurations, which coincides with the required number of simulations, amounts to ${n_\Gamma}^{n_{\mathrm{WT}}-n_{\mathrm{s}}}$ if the n_{s} downstream-most turbines run greedily. Analogously, again running the downstream-most turbine greedily, an upstream section with $n_{\mathrm{WT},u}$ turbines admits ${n_\Gamma}^{n_{\mathrm{WT},u}-1}$ possible yaw configurations and a covering section S_k with $n_{\mathrm{WT},k}$ turbines admits $n_{\Delta,k} := {n_\Gamma}^{n_{\mathrm{WT},k}-1}$.

To solve a single WFYP instance, we perform precomputations, i.e., simulation runs for all yaw configurations on all possible section configurations; recall that we include all possible ones (not only those that occur as covering sections) to preserve

flexibility, see Sect. 2.2.2, and thus, we derive the worst-case number of simulations. In addition, remembering that farm layout and wind conditions (in particular, direction and speed) are fixed for a single WFYP instance. Consequently, while for each scenario, the number of simulations is much lower than in the basic approach, preparing our approach for application in a variety of wind scenarios for a given farm will still result in a large precomputation time to run all required simulations. However, we propose to store these precomputed simulation results in a database so that data corresponding to any currently encountered scenario can be retrieved efficiently to solve the corresponding WFYP instance in order to update the yaw control.

In the following, we derive the number of combinations (section configurations and corresponding yaw configurations) and the required number of simulation runs. As there are different ways to avoid further redundant computations in specific situations, these numbers are upper bounds and might be further reduced; we will mention some examples of this aspect.

An upstream section (i.e., complete section configuration) with $n_{\mathrm{WT},u}$ turbines has $2^{n_{\mathrm{WT},u}} - 1$ non-empty section configurations. All other section configurations have fewer active turbines than $n_{\mathrm{WT},u}$, and, consequently, admit fewer possible yaw configurations (than the complete one), i.e, the number of required simulation runs is smaller than $(2^{n_{\mathrm{WT},u}} - 1)\, {n_\Gamma}^{n_{\mathrm{WT},u}-1}$. We derive the exact worst-case number of simulations needed, i.e., the total count of all yaw configurations for all possible section configurations (for $n_{\mathrm{WT},u}$ turbines) that are non-empty and have an active downstream-most turbine running greedily (cf. Sect. 2.2.3). The remaining ones can then either be inactive or active with one of n_Γ yaw offsets. Thus, to select $n \in \{0, 1, \ldots, n_{\mathrm{WT},u} - 1\}$ active turbines among these, there are $\binom{n_{\mathrm{WT},u}-1}{n}$ distinct possibilities and, for any selection of n active turbines, ${n_\Gamma}^n$ possible yaw configurations. Thus, the total number of simulations amounts to

$$n_{\mathrm{sim}} := \sum_{n=0}^{n_{\mathrm{WT},u}-1} {n_\Gamma}^n \binom{n_{\mathrm{WT},u} - 1}{n}. \tag{5}$$

In case of our Example 2.2 (3×2 farm), see Fig. 5(a), with $n_{\mathrm{WT},u} = 4$ turbines in the upstream section, the formula yields $7^0\binom{3}{0} + 7^1\binom{3}{1} + 7^2\binom{3}{2} + 7^3\binom{3}{3} = 512$ combinations (i.e., simulation runs); this also applies to the 4×2 farm, see Fig. 5(b), and all enlargements orthogonal to wind direction as they build on the same upstream section. In comparison, if all turbines are active (and the downstream-most ones still run greedily), the basic approach (full enumeration) leads to ${n_\Gamma}^3 = 7^3 = 343$ required simulations (for 3×2) and to $7^4 = 2401$ (for 4×2), which seems to be a better choice for the 3×2 farm. However, the CA already includes the possibility to deactivate any turbines, see Sect. 2.2.3. If this were included in the basic approach, we would end up with $(n_\Gamma + 1)^3 \cdot 2^3 = 8^3 \cdot 2^3 = 4096$ combinations (for 3×2) and $8^4 \cdot 2^4 = 65\,536$ (for 4×2). Thus, we expect that the CA provides a significantly higher efficiency than the basic approach for most real-world farm layouts and wind directions.

Recall that we need these precomputations for each wind condition, in particular, we focus on direction and speed, see Sect. 1.2.1. Usually, these are also discretized in a wind rose, see, e.g., the figures in Zhang et al. (2014); Fleming et al. (2016), whereas extreme speeds are summarized separately, see Gebraad et al. (2017). Analogously to the yaw offset discretization, a finer discretization is possible but questionable due to the uncertainty of incident wind conditions as discussed in Sect. 1.2.1.

2.3 Formulation of the covering approach as an IP

It remains to formalize how we can use the covering sections to solve the WFYP globally optimal. So, recall the idea to represent the farm as a set of overlapping covering sections (cf. Def. 2.4) rather than of single turbines. Instead of deciding

directly on the yaw offset of each turbine, decision variables assign a specific yaw configuration to each covering section. For the consistency of the farm covering, we require that each turbine in intersecting parts of different covering sections consistently has the same yaw offset in these. This, together with the requirement that each covering section is assigned exactly one yaw configuration, mirrors the constraint of the basic approach that each turbine can only be set at one yaw offset, cf. Eqs. (2) to (4).

2.3.1 Contributions of wind turbines located at overlaps of covering sections

Recall that in the basic WFYP approach, the objective function has coefficients (from simulations) for each turbine and yaw offset configuration. Now, we have contributions related to assigning yaw configurations (with respect to the underlying section configuration) to covering sections. To avoid multiple counting of the individual contributions of turbines located at overlaps of covering sections, which are available from the simulation results (see vector-valued function f_ω in Sect. 2.1), we consider the covering sections in order of their indices $(S_1, S_2, \ldots, S_{n_\mathrm{s}})$ and construct the objective by only adding contributions of turbines in a current covering section S_k if they were not already contained in the previous covering sections $S_1, \ldots, S_{k-1}$. Let $T_k := S_k \setminus (\cup_{m=1}^{k-1} S_m)$ denote the set of new turbines in covering section S_k; e.g., in the example from Fig. 5(a), it holds that $T_1 = \{1, 2, 4\}$, $T_2 = \{3, 5\}$, and $T_3 = \{6\}$. Then, we can express the WFYP objective value of a given yaw configuration assignment (one ℓ_k for each respective covering section S_k) with our previously-used black-box function as

$$\sum_{k=1}^{n_\mathrm{s}} \sum_{i \in T_k} f_{\omega,i,j(\ell_k)}(x(\ell_k)), \tag{6}$$

where $x(\ell_k)$ stands for the individual-turbine yaw offset settings across covering section S_k, which now depend on the yaw configuration ℓ_k given for each section S_k, and $j(\ell_k)$ is the corresponding yaw offset index of turbine i.

2.3.2 Compatibility of yaw configurations in covering sections

We will discuss the consistency of the farm covering, whereby we describe the details of the CA and lead up to the IP (8) to (11). Again, for simplicity, we assume the same set of admissible yaw offsets, say, Γ with $n_\Gamma := |\Gamma|$ for the WTs. The appropriate covering sections (and required underlying configuration sections) are defined before IP model building, recall Sect. 2.2.3.

To specify the IP model, we need additional notation. For a covering section $S_k \subseteq T$ ($k = 1, \ldots, n_\mathrm{s}$) with $n_{\mathrm{WT},k}$ turbines, we identify the yaw configurations inside S_k by indices $\ell_k = 1, 2, \ldots, n_{\Delta,k}$ with $n_{\Delta,k} := {n_\Gamma}^{n_{\mathrm{WT},k}-1}$ (as defined in Sect. 2.2.4). Let $\gamma_i(\ell_k)$ denote the yaw offset assigned to turbine $i \in S_k$ under yaw configuration ℓ_k. For consistency of the global yaw configuration as a composition of sectional yaw configurations, the yaw configurations of overlapping covering sections must match on the yaw offsets of turbines located in the respective intersection. To that end, if the yaw configuration ℓ_k was selected for S_k, then, for any $S_{\hat{k}}$ with $S_{\hat{k}} \cap S_k \neq \emptyset$ for $\hat{k} \neq k$, only a subset of yaw configurations is compatible with this selection, namely those $\ell_{\hat{k}} \in \{1, 2, \ldots, n_{\Delta,\hat{k}}\}$ for which the yaw offsets $\gamma_i(\ell_{\hat{k}}) = \gamma_i(\ell_k)$ for all WTs $i \in S_{\hat{k}} \cap S_k$. In fact, it suffices to enforce these conditions explicitly for directly adjacent pairs of covering sections (which explicitly excludes an arbitrary order), if they are numbered in ascending sequence in accordance with the downstream-most turbines (say, $1, \ldots, n_\mathrm{s}$ from left to right from behind the farm looking against the wind direction). Then, establishing consistency of the respective overlaps of S_k and S_{k+1}

Table 1. Lexicographic indexing for the $n_{\Delta,1} = 3^2 = 9$ yaw configurations for the underlying section configuration to covering section $S_1 = \{1, 2, 4\}$ in the 3×2 farm Example 2.2, cf. Fig. 5(a), with simplified $\Gamma = \{-15°, 0°, +15°\}$ (and WT 4 fixed to $0°$).

ℓ_1	1	2	3	4	5	6	7	8	9
WT 1	$-15°$	$-15°$	$-15°$	$0°$	$0°$	$0°$	$+15°$	$+15°$	$+15°$
WT 2	$-15°$	$0°$	$+15°$	$-15°$	$0°$	$+15°$	$-15°$	$0°$	$+15°$
WT 4	$0°$	$0°$	$0°$	$0°$	$0°$	$0°$	$0°$	$0°$	$0°$

Table 2. Valid yaw configurations $\tilde{L}_{k+1,\ell_k}$ of covering section S_{k+1} (depending on yaw configuration ℓ_k of the previous one S_k) compared to all possible yaw configurations L_{k+1} of S_{k+1} for the 3×2 farm from Example 2.2, Fig. 5(a), assuming admissible yaw offset offsets $\{-15°, 0°, +15°\}$ (simplified for the example) for every turbine, fixed yaw offset $0°$ for downstream-most turbines and yaw configurations being indexed in lexicographical order. For S_1, all possible yaw configurations $L_1 = \{1, 2, \ldots, 9\}$ are also valid by design, i.e., $\tilde{L}_1 = L_1$.

k	1	1	1	…	1	2	2	2	…	2
ℓ_k	1	2	3	…	9	1, 4, 7	2, 5, 8	3, 6, 9	…	21, 24, 27
L_{k+1}	$\{1,\ldots,27\}$	$\{1,\ldots,27\}$	$\{1,\ldots,27\}$	…	$\{1,\ldots,27\}$	$\{1,\ldots,9\}$	$\{1,\ldots,9\}$	$\{1,\ldots,9\}$	…	$\{1,\ldots,9\}$
$\tilde{L}_{k+1,\ell_k}$	$\{1,2,3\}$	$\{4,5,6\}$	$\{7,8,9\}$	…	$\{25,26,27\}$	$\{1\}$	$\{2\}$	$\{3\}$	…	$\{9\}$

by resorting to valid yaw configurations for S_{k+1} (defined relative to S_k and each ℓ_k), for $k = 1, \ldots, n_\mathrm{s} - 1$, is indeed enough to guarantee global consistency, which has to overcome the sequential order and is realized in Eq. (10), as by construction, for any WT $i \in S_{\hat{k}} \cap S_k$ with $\hat{k} \geq k + 2$, necessarily also WT $i \in S_{k+1}$. In addition to $L_k := \{1, 2, \ldots, n_{\Delta,k}\}$, the index set of all possible yaw configurations for S_k, we therefore also need the set of *valid* (or *compatible*) *yaw configurations* for S_{k+1} relative to S_k with $\ell_k \in L_k$, which we denote as $\tilde{L}_{k+1,\ell_k} := \{\ell_{k+1} \in L_{k+1} : \gamma_i(\ell_{k+1}) = \gamma_i(\ell_k) \text{ for all } i \in S_{k+1} \cap S_k\}$. For S_1, all possible yaw configurations in L_1 are already valid, i.e., $\tilde{L}_1 = L_1$, as S_1 has no "preceding" covering section. Tables with these dependencies, i.e., the set of valid yaw configurations $\tilde{L}_{k+1,\ell_k}$ for the current covering section (numbered as $k + 1$) in dependence of the previous one (numbered as k) and its chosen yaw configuration ℓ_k, can be computed straightforwardly.

To illustrate the notions, we use Example 2.2 (3×2 farm) again. For each covering section, marked in Fig. 5(a), we need to uniquely identify every possible yaw configuration with an index, e.g., by sorting them lexicographically with respect to the yaw offsets (in increasing order of the turbine indices); Table 1 shows an example for covering section S_1, assuming for simplicity $\{-15°, 0°, +15°\}$ as admissible yaw offsets—downstream-most turbines (4, 5, and 6) are fixed to $0°$, cf. Sect. 1.2.1.

Assuming the same lexicographic indexing to the yaw configurations for each section configuration (and thus for the covering sections), we can determine the sets of valid yaw configurations; Table 2 shows the sets $\tilde{L}_{k+1,\ell_k}$ for our simplified example. For instance, if yaw configuration $\ell_1 = 3$ was used for S_1, then only those yaw configurations for S_2 in which turbines 1 and 2 also have yaw offsets $-15°$ and $+15°$, respectively, are valid for S_2; with the used indexing, this amounts to $\tilde{L}_{2,3} = \{7, 8, 9\}$. For $\ell_2 = 7$, only yaw configuration $\ell_3 = 1$ is valid; indeed, $\tilde{L}_{3,7} = \tilde{L}_{3,1} = \tilde{L}_{3,4} = \{1\}$, as these yaw configurations for S_2 set both turbines 2 and 3 to $-15°$, for which the only compatible (and hence, valid) yaw configuration for S_3 is precisely $\ell_3 = 1$.

2.3.3 WFYP formulation as regular IP

Now, we introduce binary decision variables y_{k,ℓ_k} that encode whether covering section S_k is assigned yaw configuration ℓ_k ($y_{k,\ell_k} = 1$) or not ($y_{k,\ell_k} = 0$). Using these variables, the black-box objective function, cf. Eq. (2), can be replaced by a fully linear one once we have precomputed the simulation results for the section configurations. Indeed, the simulation results allow us to specify cost coefficients c_{k,ℓ_k} for every pair of a covering section S_k and any one of its associated yaw configurations ℓ_k; in order to avoid counting the objective contributions of turbines within intersecting parts of different covering sections multiple times, we again use a summation that only considers contributions of new turbines in a covering section, cf. Eq. (6):

$$c_{k,\ell_k} := \sum_{i \in T_k} f_{\omega,i,j(\ell_k)}(x(\ell_k)). \tag{7}$$

To achieve a globally optimal yaw configuration for the whole farm, we now have to optimize over all compatible combinations, see Sect. 2.3.2, of covering section and yaw configuration assignments (each of which has one associated coefficient c_{k,ℓ_k} and one decision variable y_{k,ℓ_k}). This yields the following integer linear program to solve the WFYP:

$$\max_{y} \quad \sum_{k=1}^{n_\mathrm{s}} \sum_{\ell_k=1}^{n_{\Delta,k}} c_{k,\ell_k}\, y_{k,\ell_k} \tag{8}$$

$$\text{s.t.} \quad \sum_{\ell_k=1}^{n_{\Delta,k}} y_{k,\ell_k} = 1 \qquad \text{for } k = 1,\dots,n_\mathrm{s}, \tag{9}$$

$$0 \le \sum_{\ell_{k+1} \in \tilde{L}_{k+1,\ell_k}} y_{k+1,\ell_{k+1}} - y_{k,\ell_k} \le 1 \qquad \text{for } k = 1,\dots,n_\mathrm{s}-1 \quad \text{and} \quad \ell_k = 1,\dots,n_{\Delta,k}, \tag{10}$$

$$y_{k,\ell_k} \in \{0,1\} \qquad \text{for } k = 1,\dots,n_\mathrm{s} \quad \text{and} \quad \ell_k = 1,\dots,n_{\Delta,k}. \tag{11}$$

Constraints (9) ensure that exactly one yaw configuration is selected for each covering section, analogously to Eq. (3). Constraints (10) ensure compatibility as they enforce the selected yaw configuration for a covering section S_{k+1} to be *valid* with respect to the yaw configuration chosen for its preceding one S_k, as described in Sect. 2.3.2: if $y_{k,\ell_k} = 1$, i.e., S_k uses yaw configuration ℓ_k, then for S_{k+1}, a yaw configuration from $\tilde{L}_{k+1,\ell_k}$ must be selected, i.e., one of the associated binary variables—and hence, their sum—must be one. If $y_{k,\ell_k} = 0$, the constraint imposes no restriction[1] with respect to $\tilde{L}_{k+1,\ell_k}$.

Finally, we emphasize that both the black-box IP (2) to (4) and the regular IP (8) to (11) are different formulations of *the same problem*, i.e., the WFYP; as such, they are equivalent—strictly speaking, this is only true if we take into account even the smallest wake-induced wind speed reduction to determine the upstream section instead of our practical assumption in Sect. 1.2.1; an example in Sect. 4 illustrates this small model inaccuracy. Nevertheless, the CA exploits the problem structure in a way that can significantly reduce the required number of simulations and enables the utilization of modern IP solvers to perform efficient implicit enumeration by branch and bound, thereby avoiding full enumeration.

[1] The upper bound in Eq. (10) is redundant: we investigated the effects for 6×3 to 9×3 farms as in series 2, see Table 6; the redundant conditions slightly increase the solving time of the IPs if Gurobi is used as solver, namely (in s) 0.25 (instead of 0.22), 1.17 (1.09), 3.79 (3.50), 10.77 (10.48), but decrease it in three out of four cases with SCIP, namely (in s) 18.57 (21.45), 154.28 (156.47), 436.67 (446.62), 3278.30 (1781.71).

3 Simulation

We obtain the simulation function output from simulation software which is interchangeable in our approach. Moreover, recall that our WFYP IPs (2) to (4) or (8) to (11) need simulation function values for different yaw offset configurations. As we control the yaw offsets, we only denote the corresponding decision variables x as simulation function arguments; the other inputs (farm layout, wind direction and speed) will be made clear in our experiments. In the following, we also introduce other conditions (and used fixed values) on which the simulation also depends.

3.1 Simulation software and parameter setup

For the farm simulations, we used the software package WinFaST[2]. This simulation framework requires a fixed farm layout. Axial induction and yaw offsets can be set time-dependent. As our focus lies on optimal yaw offsets, we leave the greedy control with respect to axial induction to the local controller. The dynamic wake model of WinFaST is based on FLORIDyn, see Gebraad and van Wingerden (2014). As FLORIDyn, it uses so-called observation points to compute local wake characteristics and wake interaction is based on Katic et al. (1987). The turbine controller in WinFaST is inspired by Jonkman et al. (2009), which is widely used for NREL 5-MW turbines, extended by the options (not used by us) to reduce the power and damp tower oscillations, each with respect to its own respective turbine. Moreover, WinFaST uses a modified version (to include yaw control and effects) of the dynamic wind turbine model by Ritter et al. (2016, 2018). The wind field in WinFaST is simulated by Veers method, see Veers (1988).

We denote the average wind speed value of the (horizontal) ambient wind field by U_{ave}. The TI is defined as $I = \sigma/U_{\mathrm{ave}}$, where σ is the associated standard deviation; it depends on the average wind speed, the roughness of the surface, the atmospheric stability, and the topography, see, e.g., (Hau, 2013, Sect. 13.4). The software WinFaST uses the same parametric model parameters for turbine and wake as in (Gebraad and van Wingerden, 2014, Table 1), that were adjusted for $8\,\mathrm{m\,s^{-1}}$ with a TI of 6%, with the exception of the air density, which is set to $1.225\,\mathrm{kg\,m^{-3}}$ as in (Jonkman et al., 2009, Appendix B.1). In our exemplary experiments, we fix the TI to 6%.

3.2 Performance indicators and simulation function

It takes a while for the wake of the upstream-most turbine(s) to reach the downstream-most one within the upstream section. Thus, we need to choose a sufficiently long *simulation time interval* $[t_{\mathrm{s}_1}, t_{\mathrm{s}_2}]$, depending on the wind speed, the TI, and upstream section layout. Moreover, for data analysis and as yaw offsets are adjusted at a fairly low rate, we are only interested in the simulation part in which the wake already influences the downstream-most turbine. Also, the wind field is equipped with turbulence and the turbines produce some, so we analyze data over an *observation time interval* $[t_{\mathrm{o}_1}, t_{\mathrm{o}_2}]$; we use it to compute the performance indicators and to define our simulation function. In practice, we round the minimal wind speed in the wind field down to $0.5\,\mathrm{m\,s^{-1}}$ (namely, 4.5, 9, and $10\,\mathrm{m\,s^{-1}}$ for speeds of 6, 11, and 12 with a TI of 6%) and simulate with this speed

[2]The MATLAB package WinFaST (Wind Farm Simulation Tool), written by Bastian Ritter and Thorsten Schlicht, is proprietary software for company-internal use at our industry partner IAV GmbH, who provided it to us for experimentation within the joint MOReNet project.

and a TI of 0% to round up the resulting propagation time in minutes to finally set a robust value for t_{o_1}, see Table 3 in Sect. 4 for examples. As we choose a duration of $10\,\mathrm{min}$ (to obtain roughly the specified wind speed on a mean at WT 3, cf. Table 3), we end up with $t_{\mathrm{o}_2} = t_{\mathrm{o}_1} + 10\,\mathrm{min}$ and $[t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] = [0, t_{\mathrm{o}_2}]$.

The performance indicators (cf. Eq. (13) later) consist of the following three outputs of WinFaST: The *power* generated by each turbine is given in the unit W as function $p_x : [t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] \to \mathbb{R}_{\geq 0}^{n_{\mathrm{WT}}}$. To compute loads we use the *velocity of the nacelle* in unit $\mathrm{m\,s^{-1}}$ in wind direction, $v_x : [t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] \to \mathbb{R}^{n_{\mathrm{WT}}}$, and the *blade pitch angle* in the unit degree, $\beta_x : [t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] \to \mathbb{R}^{n_{\mathrm{WT}}}$.

Now, we define the three performance indicators for each turbine i as averages over the observation time interval $[t_{\mathrm{o}_1}, t_{\mathrm{o}_2}]$, namely the power (output) P_i (in MW), the *tower activity* $a_i^{(\mathrm{T})}$, and the *pitch activity* $a_i^{(\mathrm{P})}$. The tower load is high when the nacelle is oscillating; therefore, we use the absolute value of the nacelle velocity v to estimate the tower load by the so-called tower activity. The pitch rate should be kept within limits because of the load of the pitch actuators; therefore, similarly, we use the absolute value of the velocity of the blade pitch angle β to estimate the load of the pitch actuators by the so-called pitch activity. The performance indicators are defined as follows:

$$P_i(x) := \frac{1}{t_{\mathrm{o}_2} - t_{\mathrm{o}_1}} \int_{t_{\mathrm{o}_1}}^{t_{\mathrm{o}_2}} 10^{-6} \, (p_x(t))_i \,\mathrm{d}t, \;\; a_i^{(\mathrm{T})}(x) := \frac{1}{t_{\mathrm{o}_2} - t_{\mathrm{o}_1}} \int_{t_{\mathrm{o}_1}}^{t_{\mathrm{o}_2}} |(v_x(t))_i| \,\mathrm{d}t, \;\; a_i^{(\mathrm{P})}(x) := \frac{1}{t_{\mathrm{o}_2} - t_{\mathrm{o}_1}} \int_{t_{\mathrm{o}_1}}^{t_{\mathrm{o}_2}} \left| \frac{\mathrm{d}}{\mathrm{d}t} (\beta_x(t))_i \right| \mathrm{d}t. \quad (12)$$

The respective units of tower and pitch activity are $\mathrm{m\,s^{-1}}$ and $^\circ\,\mathrm{s^{-1}}$ but have no physical meaning.

Finally, we define the weighted sum of these three performance indicators as the simulation function depending on the control input, i.e., the decision variables x. Recall that the dependence on yaw configurations also includes that of a turbine $i \in T$ on its own yaw offset, which can be expressed using the yaw offset index $j \in n_{\Gamma,i}$. Therefore, following the notation introduced in Sect. 2.1, we write $P_{i,j}(x)$, $a_{i,j}^{(\mathrm{T})}(x)$ and $a_{i,j}^{(\mathrm{P})}(x)$. With weights $\omega = (\omega^{(\mathrm{T})}, \omega^{(\mathrm{P})}) \in \mathbb{R}_{\geq 0}^2$ for the activity terms the entries of the simulation function, which yields the black-box function to maximize, are:

$$f_{\omega,i,j}(x) := P_{i,j}(x) - \omega^{(\mathrm{T})} \, a_{i,j}^{(\mathrm{T})}(x) - \omega^{(\mathrm{P})} \, a_{i,j}^{(\mathrm{P})}(x). \quad (13)$$

It represents our two main objectives when controlling the farm: maximizing the total power output and minimizing the turbines' mechanical load; the weights balance these typically conflicting objectives. In fact, they could even do this individually for each turbine. Moreover, for simplicity, we focus on the power output, i.e., usually $\omega^{(\mathrm{T})} = \omega^{(\mathrm{P})} = 0$ in Sect. 4. In practice, one "simulation run" consists of evaluating the (vector-valued) simulation function $f_\omega : \{0,1\}^{n_{\Gamma,1} + \cdots + n_{\Gamma,n_{\mathrm{WT}}}} \to \mathbb{R}^{n_{\mathrm{WT}}}$ (with entries of the form $f_{\omega,i,j}(x)$) once for the associated section and assignment of decision variables x.

4 Computational results

In order to obtain computational results, we show the overall process in Fig. 6 as it is a combination of CA, see Sect. 2.2, its formulation as IP, see Sect. 2.3, and simulation, see Sect. 3.

All computations were carried out on a Linux workstation with an Intel(R) Core(TM) i7-6700 CPU with $3.40\,\mathrm{GHz}$ (4 cores, 8 threads) and $32\,\mathrm{GB}$ memory. The precomputations (simulation runs) were done using MATLAB R2024b, utilizing

1. Setup: set a wind direction, speed, and TI, e.g., 6%; choose a farm layout (e.g., 6×3) and admissible yaw offsets.
2. Preparation and precomputation:
   (a) Determine the size of the upstream section (for a TI of 0% with 0° and the extremes as yaw offsets), see Sect. 2.2.1.
   (b) Set the observation and simulation time intervals with respect to the propagation time and the TI, see Sect. 3.2 and Table 3.
   (c) Determine the section configurations, i.e., all patterns of the upstream section, see Sect. 2.2.2.
   (d) Precomputation: run simulations for all yaw configurations on all possible section configurations, see Sect. 2.2.4.
   (e) Evaluate the precomputations in the observation time interval to set up the simulation function, see Sect. 3.2.
3. Determine the covering sections, i.e., the required section configurations to cover the farm, see Sect. 2.2.3.
4. Formulate the CA as an IP, see Sect. 2.3.
5. Solve the IP (e.g., with Gurobi in less than $1\,\mathrm{min}$); for comparison, simulate the entire farm with baseline and optimized yaw offsets.

Figure 6. Overall process to obtain computational results (as combination of CA, IP formulation, and simulation).

parallelization of 4 workers. The IPs resulting from our CA were solved with state-of-the-art IP solvers, namely the open-sourced academic solver SCIP $8.0.3$, utilizing the LP solver SoPlex $6.0.3$, which only supports single-thread, see Bestuzheva et al. (2021), as well as the proprietary Gurobi $10.0.0$, which can employ all threads, see Gurobi Optimization, LLC (2022).

Before we discuss the results, we briefly summarize the most important parts of the overall experimental setup. As admissible yaw offsets we choose $\gamma \in \Gamma = \{-15^\circ, -10^\circ, \dots, 15^\circ\}$ for NREL 5-MW turbines with a rotor diameter of $D = 126\,\mathrm{m}$, see (Jonkman et al., 2009, Table 1-1), in different farm layouts from 6×3 to 9×3 with a turbine spacing of $3D \times 5D$. As wind directions we use 0°, 5°, 10° or 20°, where 0° represents wind blowing from west to east. In all figures, we indicate the wind direction by a vector. As average wind speeds we consider 6, 11, and $12\,\mathrm{m\,s^{-1}}$. Deviations from this setup are made clear where they occur. Finally, we frequently compare the optimized yaw offsets against the baseline, i.e., yaw offsets of 0°.

The main computational experiments in a farm with our CA are in Sections 4.1 to 4.4: for different wind directions, for reused precomputed simulations to demonstrate flexibility, for different wind speeds, and for different yaw offset discretizations. However, we begin with some experiments to compare the WinFaST simulation with the software FLORIS, see NREL (2024), and continue with a validation of our CA on the basis of the simulation software FLORIS by comparison with serial-refine (available in FLORIS), see Fleming et al. (2022), and full enumeration (as assumptions and discretizations were employed to arrive at our IP model for the WFYP via the CA).

To that end, we consider a 3×3 farm with different TIs, wind speeds and directions and compare the baseline simulations of WinFaST and FLORIS in Table 3; see Fig. 7(a) for a visualization of part c). In detail, we use the available file gch.yaml as input in FLORIS, i.e., the Gauss-Curl-Hybrid model for wakes but set the wind shear to 0. While the results are similar at WT 3, they differ significantly at WTs 5, and 7 as WinFaST generates a wind field and uses dynamic models for turbines and wakes whereas FLORIS generates no wind field and is restricted to static models, whereby the wake model is sophisticated. Simulation is a complex topic, e.g., WinFaST simulates that WT 3 sometimes reaches $5\,\mathrm{MW}$ during observation due to turbulence in parts c) to f). A comprehensive comparison of the simulations (and which one is preferable) is out of scope of the present paper.

For the mentioned CA validation with FLORIS software simulation, we consider a 3×3 farm with a wind speed of $11\,\mathrm{m\,s^{-1}}$ and a wind direction of $20°$, i.e., the controlled WTs are 2, 3, 5, and 6, cf. the assumptions in Sect. 1.2.1. For the results we refer to Table 4 a) to c) and Fig. 7. The full enumeration takes $9.1\,\mathrm{min}$ and results in four solutions with an optimized total power output of $38.66\,\mathrm{MW}$. The serial-refine method also results in $38.66\,\mathrm{MW}$. In particular, we allow seven yaw offsets in the first phase, i.e., our usual choice of $5°$ steps, and two yaw offsets in the second phase, which practically corresponds to $2.5°$ steps, which is actually used for WT 2. This method takes only $1.07\,\mathrm{s}$. Our CA predicts a power of $38.69\,\mathrm{MW}$, but the full farm simulation with the optimal yaw offsets also results in $38.66\,\mathrm{MW}$. The CA takes $1.9\,\mathrm{min}$ for precomputation and $0.12\,\mathrm{s}$ for the IP solver SCIP, which is not directly comparable to serial-refine as we designed the CA with flexibility in mind, see Sect. 2.2.2. All in all, both serial-refine and CA have found a global solution.

The deviations of the predicted power by CA and full farm simulation (in baseline and/or optimization) are limited and are due to the size of the upstream section, which influences the accuracy of the discretized, section-based WFYP model and therefore is a compromise in terms of accuracy and run time. For example, the covering section (based on upstream section Fig. 3(d)) anchored at WT 4 includes WT 2, but WT 1 is marginally outside. An increase in size of the upstream section would ameliorate the small model inaccuracy, but also significantly increase the precomputation time. Therefore, we accept small inaccuracies in the predicted power output by CA (and run full simulation with optimized yaw offsets at the end). As the improvements over the baseline are significantly larger than the deviations from full farm simulation, the optimal solutions of our CA will likely still be optimal for a model using increased upstream sections, though suboptimality is technically possible. The IP solver run times themselves will indicate that handling larger upstream sections should not pose an issue.

We would like to present a case where our CA method is more advantageous in terms of accuracy than the fascinatingly fast serial-refine method. In theory, there could be such a case as our CA is equivalent to full enumeration for sufficiently large upstream sections, see Sect. 2.3.3. However, we have not yet found such an example.

For the remaining paper we show results of our CA approach using WinFaST simulation for computations and visualizing the results with software FLORIS. For the example above, we receive other optimized yaw offsets, see Table 4 d) and Fig. 7(b), where the precomputation takes $4.5\mathrm{h}$ and the IP solver SCIP $0.11\,\mathrm{s}$. These yaw offsets as input in FLORIS result in $38.27\,\mathrm{MW}$, which is below the baseline with FLORIS, and demonstrates the dependency of the solution on the selected simulation. In addition, we use the introduced weights $\omega = (\omega^{(\mathrm{T})}, \omega^{(\mathrm{P})}) = (100, 10)$ once to take into account the tower activity and pitch activity in optimization, cf. Eq. (13), see Table 4 e). A detailed comparison to d) with our default $\omega = (0, 0)$ shows that both tower and pitch activity are decreased, namely to 0.1105 (from 0.1108) and 0.3865 (0.4070) for the price that the power is also decreased (to 33.03 from $33.05\,\mathrm{MW}$). (To complete the impression, the activities in baseline are 0.1128 and 0.5347.)

4.1 Wind farm yaw offset optimization under different wind directions

In series 1, we consider a 6×3 farm with a wind speed of $11\,\mathrm{m\,s^{-1}}$ and wind directions of $0°$, $5°$, $10°$ and $20°$. The respective results are presented in Table 5, see also Fig. 9(b) for case 4. In all cases, the improvement of the total power output is between 2% and 17%. In most of these cases, the optimal yaw offsets exhaust the given limits of $\pm 15°$. In case 4 (i.e., wind direction

Table 3. Data and results for baseline simulations of a 3×3 farm to compare WinFaST and FLORIS (series 0, case 0). Both observation time interval $[t_{o_1}, t_{o_2}]$ and mean wind speed at WT 3 are related to WinFaST; we set t_{o_1} and $t_{o_2} = t_{o_1} + 10\,\mathrm{min}$ as discussed in Sect. 3.2.

	wind farm			obs. time	mean speed	power output with WinFaST			power output with FLORIS		
part	TI (in %)	speed (in $\mathrm{m\,s^{-1}}$)	direction (in °)	begin t_{o_1} (in min)	at WT 3 (in $\mathrm{m\,s^{-1}}$)	WT 3 (in MW)	WT 5 (in MW)	WT 7 (in MW)	WT 3 (in MW)	WT 5 (in MW)	WT 7 (in MW)
a)	0	11	0	7	11.00	4.5409	0.9110	0.3466	4.5625	0.8336	0.3157
b)	0	11	20	5	11.00	4.5409	3.3658	1.9163	4.5625	4.4762	3.1604
c)	6	11	20	6	11.01	4.4374	3.4029	1.9241	4.5625	4.4232	3.3763
d)	6	11	0	9	10.89	4.3617	1.0105	0.3704	4.5625	1.2536	1.4189
e)	6	11	5	7	11.13	4.4890	1.8419	1.1378	4.5625	2.4564	2.6977
f)	6	11	10	6	11.34	4.6021	3.3156	2.2861	4.5625	4.0900	3.9583
g)	6	6	20	12	6.07	0.7706	0.5335	0.2698	0.7376	0.7076	0.5150
h)	6	12	20	5	12.57	4.9922	4.8839	3.9316	5.0000	5.0000	4.7465

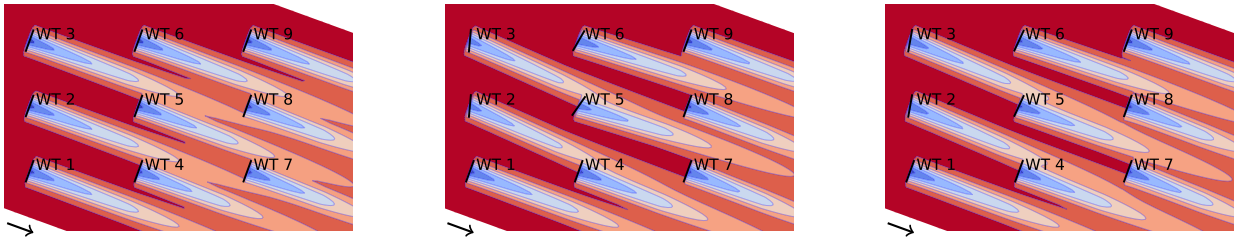
Table 4. Data and results for a 3×3 farm, a wind speed of $11\,\mathrm{m\,s^{-1}}$ (TI of 6%), and direction of $20°$ (series 0, case 1) to compare optimization methods and simulations. In cases of CA we add the run times of precomputations and IP solver SCIP. In part e) we take into account the tower activity (weighted by 100) and the pitch activity (weighted by 10), cf. Eq. 13, (marked with "(w)").

				optimal yaw offsets (in °) (of controlled WTs)				total power output		
part	simulation	opt. method	run time	WT 2	WT 3	WT 5	WT 6	baseline (in MW)	optimized (in MW)	improvement rel. (in %)
a)	FLORIS	full enumeration	9.1 min	5 or 10	10	−5	0 or −5	38.42	38.66	0.62
b)	FLORIS	serial-refine	1.07 s	7.5	10	−5	−5	38.42	38.66	0.62
c)	FLORIS	covering approach	1.9 min + 0.12 s	10	10	−5	−5	38.42	38.66	0.62
d)	WinFaST	covering approach	4.5 h + 0.11 s	15	15	−15	−10	32.64	33.05	1.25
e)	WinFaST	covering approach (w)	reuse +0.12 s	15	15	−15	−15	32.64	33.03	1.19

of $20°$), the distances between the turbines in the downstream direction are already comparatively high and consequently, the wake influence comparatively low; therefore, the improvement is 2% here, but larger in the other cases.

The main part of the overall run time is the precomputation time, e.g., in series 1 about 1 to 5 h. The IPs in our CA then are all solved in well below 1 s by Gurobi, and still within at most 19 s by SCIP. We remind the reader that the precomputations were designed with flexibility in mind, see Sect. 2.2.2, i.e., they can be reused for many cases, see our series 2 for a demonstration. Hence, for the actual optimization process used for control updates, only the IP solving times are relevant, which turned out to be so small that we can speak of real-time capable optimization (cf. Sect. 1.2.2). Finally, we see that the suggested database of precomputated simulation results, see Sect. 2.2.4, is time-consuming to build but enables significant gains through optimization.

The main influence on the precomputation time is the turbines' number in the upstream section; the impact of the allowed number of yaw offsets is smaller, cf. Sect. 4.4. The specific upstream sections in series 1 are depicted in Fig. 3; those of cases 1



(a) Baseline: 32.64 MW with WinFaST, 38.42 MW with FLORIS.

(b) Optimized with covering approach with WinFaST: 33.05 MW.

(c) Optimized with covering approach with FLORIS: 38.66 MW.

Figure 7. Baseline and optimization results for comparison (visualized with FLORIS). Subfigure captions specify the farm's power output.

Table 5. Data and results for a 6×3 farm, wind speed of $11\,\mathrm{m\,s^{-1}}$ and wind directions of $0°$, $5°$, $10°$ and $20°$ (series 1). In all cases, the IP optimality gap is 0.00%, i.e., all instances were solved to global optimality. Detailed results of case 4 are in Fig. 9(b).

	wind farm			IP solver				total power output		
case	wind direction (in °)	# covering sections	precomputation time (in h)	# variables	# constraints	solving time (in s) SCIP	solving time (in s) Gurobi	baseline (in MW)	optimized (in MW)	improvement rel. (in %)
1	0	6	0.7	294	496	0.04	0.01	35.84	38.66	7.85
2	5	6	0.7	294	496	0.04	0.01	42.14	49.49	17.45
3	10	6	4.6	1764	3436	0.54	0.23	58.07	63.68	9.67
4	20	8	4.5	1430	2866	18.57	0.25	69.04	70.26	1.76

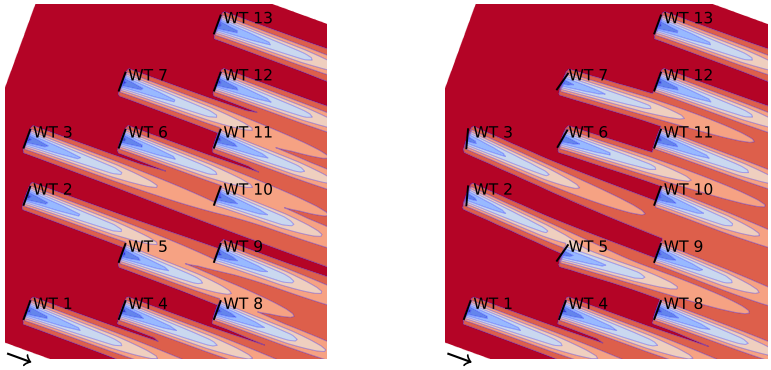
and 2 (i.e., wind directions $0°$ and $5°$) include three turbines and yield precomputation times of about an hour (cf. Table 5), whereas those of cases 3 and 4 (i.e., $10°$ and $20°$) include four turbines and take about $5\,\mathrm{h}$. Four turbines (with seven possible yaw offsets) result in 512 yaw configurations (i.e., simulations), see the example in Sect. 2.2 and Eq. (5).

4.2 Scalability and flexibility of the covering approach to solve the WFYP

In series 2, we demonstrate that reusing precomputed simulation results (for section configurations, see Sect. 2.2), namely series 1, case 4 (case 1.4), see Table 5, enables our CA to enlarge farms orthogonal to the wind direction (cases 2.1 to 2.4) and to handle cases with deactivated turbines (case 2.5). In particular, we use a wind direction of $20°$ and a speed of $11\,\mathrm{m\,s^{-1}}$ to reuse case 1.4 precomputations (originally for a 6×3 farm) for 7×3 to 9×3 and monitor the IP solvers' workload, see Table 6. The optimization consistently improves the farm's total power output by roughly 2%. The impact of the farm size on the IP solving time reflects the theoretical complexity of the WFYP (see Prop. 2.1) in practice: the solver SCIP quickly takes a long time—it could be stopped earlier if a duality gap was accepted; Gurobi is faster by orders of magnitude remaining well below one minute in all cases considered here, despite an (apparently exponential) increase in run time with the farm size. This illustrates the practical scalability of our CA and, in light of the low yaw sampling rate, its suitability for real-time WFYP

Table 6. Data and results to illustrate scalability and complexity of the WFYP (series 2). All cases have a wind direction of 20° and a speed of $11\,\mathrm{m\,s^{-1}}$. In all cases, the IP optimality gap is 0.00%. (Case 2.1 repeats case 1.4.) We estimate the predicted run time of full enumeration by $n_\Gamma{}^{n_\mathrm{WT}-n_\mathrm{s}} \cdot \tau \cdot \frac{1}{4}$, where τ is the run time of one simulation and $\frac{1}{4}$ is for parallelization effect, see Sect. 2.2.4 for number of combinations.

	wind farm		predicted run time		IP solver				total power output		
case	layout	# covering sections	of full enumeration (in min)	(in y)	# variables	# constraints	solving time (in s) SCIP	Gurobi	baseline (in MW)	optimized (in MW)	improvement rel. (in %)
1	6×3	8	$7^{18-8} \cdot 6.7 \cdot \frac{1}{4}$	$9.0 \cdot 10^2$	1430	2866	18.57	0.25	69.04	70.26	1.76
2	7×3	9	$7^{21-9} \cdot 7.7 \cdot \frac{1}{4}$	$5.1 \cdot 10^4$	1773	3553	154.28	1.17	73.29	74.48	1.63
3	8×3	10	$7^{24-10} \cdot 9.1 \cdot \frac{1}{4}$	$2.7 \cdot 10^6$	2216	4240	436.67	3.79	86.94	88.58	1.89
4	9×3	11	$7^{27-11} \cdot 10.3 \cdot \frac{1}{4}$	$1.6 \cdot 10^8$	2459	4927	3278.30	10.77	98.15	100.15	2.04



(a) Baseline: $54.51\,\mathrm{MW}$. (b) Optimized: $55.57\,\mathrm{MW}$.

Figure 8. Detailed results for a 6×3 farm in which some turbines are inactive (case 2.5 with wind speed of $11\,\mathrm{m\,s^{-1}}$ and direction of 20°). Subfigure captions specify yaw control (baseline or optimized) and the resulting total power output of the farm.

optimization even for farms with more turbines than explored here. Moreover, we demonstrate the curse of dimensionality of full enumeration via predicted run times, namely in the order of 10^2 to $10^8\,\mathrm{y}$, instead of the $4.5\,\mathrm{h}$ precomputation (cf. Table 5).

Furthermore, we handle the case 2.5 with a mix of active and inactive (or not existing) turbines to show our method's practical flexibility, e.g., to react to a shutdown for maintenance. Again, we reuse precomputations of case 1.4, cf. Table 5, but now in the setting depicted in Fig. 8, i.e., WTs $\{2, 5, 6, 9, 12\}$ are inactive. The WFYP optimization still yields about 2% improvement in generated power, namely 1.95%, while the "thinning out" of the farm leads to notably shorter IP solving times: $0.02\,\mathrm{s}$ (instead of $18.57\,\mathrm{s}$) with SCIP and below $0.01\,\mathrm{s}$ ($0.25\,\mathrm{s}$) using Gurobi.

4.3 Impact of different wind speeds

In series 3, we again consider a 6×3 farm and a wind direction of 20°, and evaluate WFYP solutions for different wind speeds. The baseline and optimization results of 6, 11 and $12\,\mathrm{m\,s^{-1}}$ are reported in Table 7 ($11\,\mathrm{m\,s^{-1}}$ repeats case 1.4) and Fig. 9: the

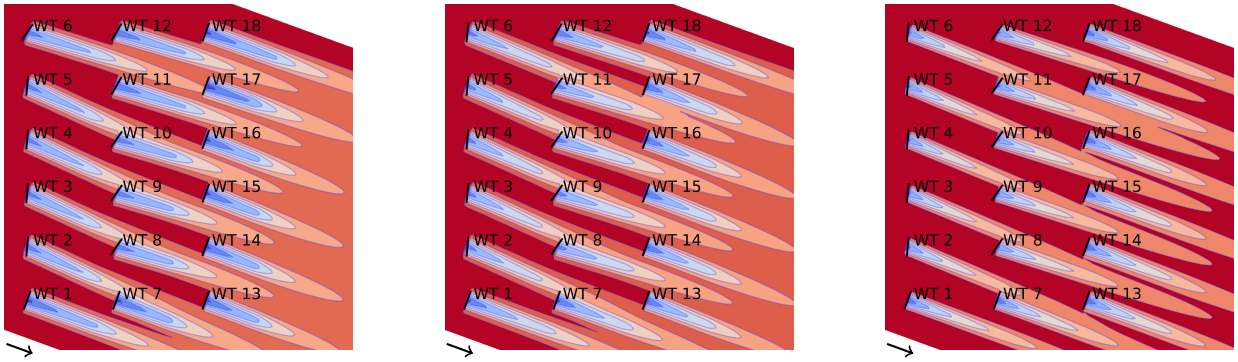
Table 7. Data and results to illustrate the impact of wind speeds (series 3). All cases have a 6×3 farm layout and a wind direction of 20° in common. In all cases, the IP optimality gap is 0.00%. Detailed results are in Fig. 9. (Case 3.2 repeats case 1.4.)

	wind farm		IP solver		total power output		
			solving time (in s)				
case	wind speed U_{ave} (in $\mathrm{m\,s^{-1}}$)	precomputation time (in h)	SCIP	Gurobi	baseline (in MW)	optimized (in MW)	improvement rel. (in %)
1	6	4.6	46.28	0.24	10.16	10.37	2.07
2	11	4.5	18.57	0.25	69.04	70.26	1.76
3	12	4.5	8.10	0.22	82.75	84.52	2.14

optimization increases the farm's total power output by about 2%; the maximum of the precomputation time is about $5\,\mathrm{h}$ and of the IP solving time about $46\,\mathrm{s}$ (SCIP) or under $1\,\mathrm{s}$ (Gurobi). Finally, further experiments provide speeds of $\{6, 7, \ldots, 15\}\,\mathrm{m\,s^{-1}}$: the improvement of the total power is between 0.90% and 2.54% for 6 to $13\,\mathrm{m\,s^{-1}}$, 0.06% for $14\mathrm{m\,s^{-1}}$, and 0.00% for $15\mathrm{m\,s^{-1}}$ (as the baseline already reaches the maximum of $90\,\mathrm{MW}$). For 6 to $14\,\mathrm{m\,s^{-1}}$ we obtain yaw offsets of the controlled WTs with a similar power output effect, namely $\pm 15^\circ$ for WTs 2 to 6 and -10° or -15° for 8 to 12. This suggests several approaches: the avoidance of many precomputations by identifying the wind speeds' "tipping points" for each wind direction, i.e., speeds that change the optimal yaw offsets (significantly with regard to the total power), the restricting of admissible yaw offsets around the found values, or the refinement of yaw offsets around the found values (similar to the serial-refine method, see Fleming et al. (2022)). Further, we observe that turbines in a row (apart from those at the farm borders) appear to typically have identical optimal yaw offsets (in the same experiment), which is likely due to the grid layout, whereby on the one hand we also have to acknowledge that our number of admissible yaw offsets is not large, on the other hand we refer to Sect. 4.4 for the effect and question of usefulness of a finer discretization. In addition, the optimal yaw offsets of all controlled WTs differ from 0°: on the one hand this may provide opportunities to reduce the running times of both precomputations and IP solving, on the other hand it might be different for other wind directions, although there are already relatively few wake interactions for the wind direction of 20° (which is reflected in the relatively small potential for total power improvement). Finally, the speed of $15\,\mathrm{m\,s^{-1}}$ is the end of interest from total power optimization perspective for the present example as the improvement is 0.00%. Nevertheless, optimization can still be meaningful if mechanical loads are included: in a further experiment, we weight the tower activity by $\omega^{(\mathrm{T})} = 100$ and the pitch activity by $\omega^{(\mathrm{P})}$, cf. Eq. (13). This results in similar optimal yaw offsets as for 6 to $14\,\mathrm{m\,s^{-1}}$, namely $\pm 15^\circ$ for WTs $\{2, \ldots, 5, 8, \ldots, 12\}$ and 10° for WT 6. The total power remains at $90\,\mathrm{MW}$, whereas the tower activity decreases to 0.4017 (from 0.4096 in baseline) but the less heavily weighted pitch activity increases to 4.2175 (from 4.1707).

4.4 Modifying the yaw offset discretization in terms of range and fineness

In series 4, we deviate from the yaw offset discretization ($[-15^\circ, 15^\circ]$ in 5° steps, i.e., seven offsets) used so far, see Table 8 for the results. In case 4.1, we use $[-40^\circ, 40^\circ]$ in 5° steps, i.e., 17 offsets, for the setup as in case 1.2 (with which we compare, cf. Table 5). This increases the precomputation time to $2.6\,\mathrm{h}$ (compared to $0.7\,\mathrm{h}$). As optimal result, all controlled turbines are set



(a) $6\,\mathrm{m\,s^{-1}}$, optimized: $10.37\,\mathrm{MW}$. (b) $11\,\mathrm{m\,s^{-1}}$, optimized: $70.26\,\mathrm{MW}$. (c) $12\,\mathrm{m\,s^{-1}}$, optimized: $84.52\,\mathrm{MW}$.

Figure 9. Optimization results for series 3 as shown in Table 7. Subfigure captions specify the wind speed and the farm's total power output.

Table 8. Data and results if yaw offset range or discretization are changed (series 4). All cases consider a 6×3 farm with a wind speed of $11\,\mathrm{m\,s^{-1}}$. In all cases, the IP optimality gap is 0.00%.

	wind farm				IP solver				total power output		
case	wind direction	allowed yaw offsets range	steps	precomputation time	# variables	# constraints	solving time SCIP	solving time Gurobi	baseline (in MW)	optimized (in MW)	improvement rel. (in %)
1	5°	$[-40°, 40°]$	5.0°	2.6 h	1734	2896	0.82 s	0.76 s	42.14	53.81	27.68
2	20°	$[-15°, 15°]$	2.5°	23.0 h	8972	17950	8.32 h	29.56 s	69.04	70.29	1.80

to $30°$, which enables a significantly increased power output of the downstream-most turbines (with $0°$), i.e., 2.9 to $3.1\,\mathrm{MW}$ (from 1.7 to $1.9\,\mathrm{MW}$). The total farm output is increased to $53.81\,\mathrm{MW}$ (from $49.49\,\mathrm{MW}$). The non-use of the new limits of $\pm 40°$ shows that at the extreme yaw offset settings, the power loss at a turbine would exceed the gain at turbines downstream.

In case 4.2, we reconsider $[-15°, 15°]$ but with $2.5°$ steps, i.e., 13 offsets, for the setup as in case 1.4 (with which we compare, cf. Table 5), which is particularly interesting for finer discretization as not all turbines attain extreme yaw offsets $\pm 15°$. Indeed, optimal yaw offsets of WTs 8 to 12 are set to $-12.5°$ (from $-15°$ or $-10°$ for WT 12), whereas WTs 2 to 6 remain at $15°$ (and others at $0°$). The longer precomputation time of $23.0\,\mathrm{h}$ ($4.5\,\mathrm{h}$) is theoretically worthwhile as the total power output is increased further to $70.29\,\mathrm{MW}$ (from $70.26\,\mathrm{MW}$). In practice, it is unnecessary to use arbitrarily fine discretizations due to the uncertainty of incident wind conditions, see Stanley et al. (2022) (referenced in Sect. 1.2.1).

5 Concluding remarks

We formulated the wind farm yaw problem mathematically, established its computational complexity and developed a covering approach, which exploits that the farm can be covered by patterns based on a smaller, precomputable so-called upstream section, in the form of integer programming to solve it faster. Building on a number of simulation results that can be precomputed at any

time before the need for yaw control arises, the method is efficient in practice in spite of the problem's strong $\mathcal{NP}$-hardness and inapproximability. In particular, we fully expect even very long precomputation times (e.g., months) to be acceptable as the simulations can easily be run for various wind scenarios while the farm is not yet operational. Given the envisioned database, our CA efficiently delivers optimal yaw control using a state-of-the-art IP solver like Gurobi. The solution is even globally optimal under some mild assumptions as discussed in Sect. 1.2.1, like discretized yaw offsets, chosen size of so-called upstream section, and homogeneous layout structure. In addition, it enables tackling even farms with many turbines. We demonstrated the performance of our approach with several proof-of-concept examples that illustrate its effectiveness, flexibility and scalability, particularly through the reuse of precomputations if we enlarge the farm orthogonally to the wind direction or deactivate turbines. On the other hand, the enlargement in wind direction increases the upstream section and therefore the number of turbines inside, which mainly increases the precomputation time. As our CA is a superordinate model, the simulation is interchangeable, e.g., to use FLORIS simulation as shown that does not utilize dynamic models (for wakes and turbines). Finally, it might be helpful to solve the WFYP for the associated farms by our CA for a variety of wind directions and speeds to recognize structures whose exploitation reduces the computational effort in precomputation or simplifies the WFYP itself. Solving the WFYP for the associated farms by CA in all considered wind scenarios should provide knowledge to reduce the computational effort.

Code and data availability. For simulation, we used the MATLAB software package WinFaST. This company-internal software is not publicly available, but based on known methods, as described in Sect. 3.1. As our own optimization framework is presently entwined with WinFaST and hence not a stand-alone program, we have not made it publicly available at this time. Nevertheless, in Sect. 2, we provide a detailed description of the problem, the novel CA, integration/utilization of simulation results and the WFYP formulation as an IP. As data to supplement the article, we provide the IPs (`lp`-files) and corresponding solver log files for each case of our series of experiments. The data is available at Zenodo: https://doi.org/10.5281/zenodo.14900916.

Appendix A: Complexity of the wind farm yaw problem

This section addresses the computational complexity of the wind farm yaw problem from a theoretical viewpoint; we assume a basic knowledge of mathematical complexity theory and refer to Garey and Johnson (1979) for a detailed introduction. Using the basic black-box IP formulation (2) to (4) of the WFYP (cf. Sect. 2.1), we show that the WFYP is strongly $\mathcal{NP}$-hard (Theorem A.3) and even hard to approximate (Corollary A.4). These two results together yield Proposition 2.1 as stated in Sect. 2.1. We use the well-known strongly $\mathcal{NP}$-complete Hamiltonian Circuit (HC) problem, see, e.g., (Garey and Johnson, 1979, problem GT37), for our proof.

Definition A.1 (Hamiltonian Circuit Problem (HCP))**.** Let an undirected graph $G = (V, E)$ on n vertices, $V := \{v_1, \ldots, v_n\}$, be given. Does G contain a Hamiltonian circuit, i.e., a subset of edges $\mathcal{H} = \big\{\{v_{\pi(1)}, v_{\pi(2)}\}, \{v_{\pi(2)}, v_{\pi(3)}\}, \ldots, \{v_{\pi(n-1)}, v_{\pi(n)}\}, \{v_{\pi(n)}, v_{\pi(1)}\}\big\} \subset E$ for a permutation $\langle v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)}\rangle$ of V?

For clarity, we also explicitly state the decision version of the WFYP problem.

Definition A.2 (WFYP decision version (WFYP-DEC))**.** Let the index set T of wind turbines in a farm, its layout $\mathcal{L}$, the index set $\Gamma_i := \{1, \dots, n_{\Gamma,i}\}$ of admissible yaw offsets for each turbine $i \in T$, the WFYP objective function f_ω^Σ as defined for its (black-box) IP formulation, and a number $F \in \mathbb{R}$ be given. Does there exist a feasible yaw configuration for the given farm (i.e., exactly one yaw offset per turbine) such that, for the associated binary yaw offset assignment vector x, $f_\omega^\Sigma(x) \geq F$?

Recall that the assignment vector $x \in \{0,1\}^{n_{\Gamma 1}+\dots+n_{\Gamma n_{\mathrm{WT}}}}$, with the entry in position $\sum_{\ell=1}^{i-1} n_{\Gamma\ell} + j$ being denoted by $x_{i,j}$ and having value 1 if and only if the i-th turbine is set to the j-th yaw offset from among the respective admissible set Γ_i. In the following, we will also use some additional notation: we denote the so-called triangular numbers by $\Delta_n := \binom{n+1}{2} = \frac{n^2+n}{2}$, abbreviate the set of the first N triangular numbers as $\Delta(N) := \{\Delta_1, \Delta_2, \dots, \Delta_N\}$, and write $[n] := \{1, 2, \dots, n\}$ for $n \in \mathbb{N}$.

We are now prepared to prove the $\mathcal{NP}$-hardness and inapproximability of the WFYP(-DEC).

Theorem A.3. *The WFYP is strongly $\mathcal{NP}$-hard.*

Proof. We show hardness for the decision version WFYP-DEC, which directly implies hardness for the optimization version (WFYP, as defined in Sect. 2), cf. Garey and Johnson (1979). To that end, we reduce from the strongly $\mathcal{NP}$-complete HCP.

Let $G = (V, E)$ be an arbitrary HCP instance, and denote $n = |V|$. We can assume w.l.o.g. that $n \geq 2$, every vertex has degree at least 2, and that G consists of a single connected component (otherwise, the answer to the HCP is trivially "no"). We construct an instance $(T, \mathcal{L}, \{\Gamma_i\}_{i\in T}, f_\omega^\Sigma, F)$ of the WFYP-DEC as follows:

We set the number of turbines to $n_{\mathrm{WT}} := \Delta_n + 1$ and identify the turbines by their index, i.e., $T := [\Delta_n + 1]$. The turbines are arranged in a triangle-like layout $\mathcal{L}$ (on a regular grid)[3] defined by the following "row" sets:

$$
\begin{aligned}
&R_1 := \{\Delta_1\} = \{1\}, && R_2 := \{\Delta_2, 2\}, && R_3 := \{\Delta_3, 5, 4\},\\
&\dots, && R_n := \{\Delta_n, \Delta_n - 1, \dots, \Delta_{n-1} + 1\}, && R_{n+1} := \{\Delta_n + 1\}.
\end{aligned}
$$

Furthermore, each turbine $i \in T$ is given the same set of admissible yaw offsets $\Gamma_i := \Gamma := [n]$, so $n_{\Gamma,i} = |\Gamma_i| = n$ for all $i \in T$. Finally, we set $F := n$, and define the terms $f_{\omega,i,j}(x)$ of the simulation function $f_\omega^\Sigma(x) := \sum_{i=1}^{n_{\mathrm{WT}}} \sum_{j=1}^{n_{\Gamma,i}} f_{\omega,i,j}(x)\, x_{i,j} = \sum_{i=1}^{\Delta_n+1} \sum_{j=1}^{n} f_{\omega,i,j}(x)\, x_{i,j}$ as

$$
f_{\omega,i,j}(x) := \begin{cases}
0 & \text{if } i = 1 (= \Delta_1) \text{ and } j \in [n],\\
1 & \text{if, for some } k \in [n],\ i = \Delta_k > 1,\ x_{\Delta_{k-1},\ell} = 1 \text{ for some } \ell \text{ such that } \{v_\ell, v_j\} \in E,\\
 & \quad \text{and } j \in [n] \setminus \left\{q \in [n] : \textstyle\sum_{s\in R_{k-1}} x_{s,q} > 0\right\}\\
0 & \text{if, for some } k \in [n],\ i \in R_k \setminus \big(\Delta(n) \cup \{\Delta_n + 1\}\big) \text{ and } x_{i-k+1,j} = 1,\\
1 & \text{if } i = \Delta_n + 1,\ x_{\Delta_{n-1}+1,j} = 1, \text{ and } \{v_j, v_\ell\} \in E \text{ for } j \text{ such that } x_{\Delta_n,\ell} = 1,\\
1 - n & \text{otherwise.}
\end{cases} \tag{A1}
$$

[3] In fact, the precise layout does not matter, since all implications regarding the resulting wake influences are "hidden" in the black-box function (i.e., in the practical application, handled within the simulation framework); the same holds for the exogenously given (arbitrary but fixed) wind speed and direction.

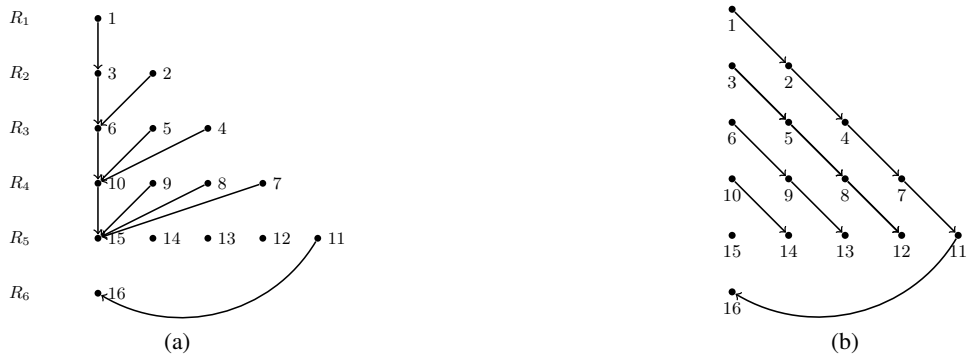


Figure A1. Visualization of black-box function dependencies in the reduction from HCP to WFYP-DEC, exemplified for graph with $n = 5$ nodes. Arcs in (a) represent the dependencies in the constructed farm (i.e., which turbines bear influence on which others) for the first two cases in Eq. (A1), while arcs in (b) represent those for the remaining cases in Eq. (A1). Note that actual function values depend on input x.

For a (feasible) overall yaw configuration of the farm as determined by x, the components $f_{\omega,i,j}(x)$ of $f^{\Sigma}_{\omega}(x)$ specify the objective contribution (or profit, for short) incurred by turbine i using the yaw offset (indexed by) j. Specifically, the first case in Eq. (A1) defines a zero profit for any arbitrary yaw offset assignment to the first turbine. The second case then yields a profit of 1 if a turbine that corresponds to a triangular number $\Delta_k > 1$ has been assigned (according to the input yaw configuration x) a yaw offset j that was not chosen for any turbine in the previous row (set R_{k-1}) and is such that for the yaw offset ℓ chosen for turbine Δ_{k-1}, the edge $\{v_\ell, v_j\}$ exists in G. (For example, supposing turbines Δ_1 and Δ_3 are assigned yaw offsets $\ell \in [n]$ and j, respectively, then the setting for turbine Δ_3 yields a profit of 1 only if $\{v_\ell, v_j\} \in E$ and the j-th yaw offset was not selected for any turbine in the previous row, which in this case translates to $j \neq \ell$.) The third case means that using yaw offset j for any turbine $i \in R_k \setminus (\Delta(n) \cup \{\Delta_n + 1\}) = \{\Delta_{k-1} + 1, \ldots, \Delta_k - 1\}$ (with respect to some $k \in [n]$) incurs zero profit if this yaw offset j was used at turbine $i - k + 1$ (which belongs to row set R_{k-1}, since $\Delta_{k-2} + 1 \leq i - k + 1 \leq \Delta_{k-1}$). The penultimate case yields a unit profit in the special case that the turbine is $i = \Delta_n + 1$ and the assigned yaw offset j is also used by turbine $\Delta_{n-1} + 1$, provided that the edge $\{v_j, v_\ell\} \in E$ for ℓ being the yaw offset assigned to turbine Δ_n. Finally, the last case sets the function value to $1 - n$ for all other configurations. Fig. A1 illustrates the dependency structure of the function.

This completes the construction of a WFYP-DEC instance $(T, \mathcal{L}, \{\Gamma_i\}_{i \in T}, f^{\Sigma}_{\omega}, F)$ from the input HCP instance G. Note that the reduction (dimensions, all arithmetic operations and constructed numbers) clearly requires only polynomial time and space with respect to the size of the input; in particular, the objective function can be evaluated in $\mathcal{O}(n^3)$, since $|T| = n_{\mathrm{WT}} = \Delta_n + 1 \leq n^2$ and $n_{\Gamma_i} = n$ for all $i \in T$. (In fact, it can easily be seen that our reduction allows the "strongly" part of $\mathcal{NP}$-hardness to carry over from the HCP, cf. Garey and Johnson (1979).)

It remains to show that the given graph contains a Hamiltonian circuit $\mathcal{R}$ if and only if the constructed WFYP-DEC instance $(T, \mathcal{L}, \{\Gamma_i\}_{i \in T}, f^{\Sigma}_{\omega}, F)$ admits a solution x with objective value $f^{\Sigma}_{\omega}(x) \geq F = n$. To that end, first assume that $\hat{x} \in \{0,1\}^{(\Delta_n + 1)n}$ is a feasible solution for WFYP-DEC (so every turbine is assigned exactly one yaw offset) with objective value $f^{\Sigma}_{\omega}(\hat{x}) \geq F$. Since by construction, only n turbine yaw settings can possibly incur a profit of 1 each (and all others at most zero), $f^{\Sigma}_{\omega}(\hat{x}) = F = n$ does in fact hold, which also implies that the last case in the function definition (A1) never occurs (since

otherwise, $f_\omega^\Sigma(\hat{x}) \leq 1 - n + n = 1 < n$ would hold—a contradiction). In particular, tracing the functional dependencies with regard to which yaw offset assignments incur which costs for subsequent turbines (in the "cascading" row sets), we can conclude that no yaw offset is chosen twice among the turbines $1, \Delta_2, \Delta_3, \ldots, \Delta_n$. Moreover, due to the first two cases in Eq. (A1) (and since $\hat{x}$ represents a feasible yaw configuration), and because every turbine has the same set of n admissible yaw offsets, it holds that each yaw offset is chosen exactly once for this set $\{1, \Delta_2, \Delta_3, \ldots, \Delta_n\}$ of turbines. Furthermore, note that by definition, any yaw setting j for turbines $i = \Delta_{k-1} + 1$, $k = 2, \ldots, n$, incurs a profit of either $1 - n$ or 0, but that since $f_\omega^\Sigma(\hat{x}) = n$, the respective settings prescribed by $\hat{x}$ in fact all yield zero profit. Thus, these costs necessarily arise from the third case in Eq. (A1), which means that turbines $\Delta_1 + 1, \Delta_2 + 1, \ldots, \Delta_{n-1} + 1$ all have the same yaw offset as turbine 1. Consequently, by the fourth case in the definition, the yaw configurations chosen for turbines $\Delta_1 = 1$ and $\Delta_n + 1$ are also identical.

We can now construct a Hamiltonian circuit in G from this WFYP solution $\hat{x}$: starting at vertex $v_{p_1} \in V$, where $p_1 \in [n]$ is the yaw offset chosen for turbine $\Delta_1 = 1$, we visit the other $n - 1$ nodes in the order prescribed by the yaw offsets selected for the turbines $\Delta_1, \ldots, \Delta_n$, and finally moving from the last node back to v_{p_1}. Indeed, this traversal produces the tour $\hat{\mathcal{R}} = \{\{v_{p_1}, v_{p_2}\}, \{v_{p_2}, v_{p_3}\}, \ldots, \{v_{p_{n-1}}, v_{p_n}\}, \{v_{p_n}, v_{p_1}\}\}$, which shows that a "yes" answer for the constructed WFYP-DEC instance yields a "yes" answer for the original HCP instance.

For the converse direction, let a circuit $\hat{\mathcal{R}} = \{\{v_{\pi(1)}, v_{\pi(2)}\}, \ldots, \{v_{\pi(n-1)}, v_{\pi(n)}\}, \{v_{\pi(n)}, v_{\pi(1)}\}\}$ be a "yes" certificate for the given HCP instance G. Then, we can derive a solution $\hat{x}$ with cost $f_\omega^\Sigma(\hat{x}) = n = F$ for the constructed WFYP-DEC instance from $\hat{\mathcal{R}}$ as follows: for turbines 1, Δ_2, Δ_3, …, Δ_n, Δ_n+1, we respectively select the yaw offsets corresponding to the indices of the vertices in the order prescribed by the tour $\hat{\mathcal{R}}$, starting (and ending) at $v_{p_1} = v_{\pi(q)}$ for some $q \in [n]$, i.e., we set $\hat{x}_{1,\pi(q+1)} = \hat{x}_{\Delta_2,\pi(q+2)} = \cdots = \hat{x}_{\Delta_n,\pi(q+n)} = \hat{x}_{\Delta_n+1,\pi(q)} = 1$ (yielding total profit $0 + n \cdot 1 = n$) while the entries corresponding to these turbines and the respective remaining yaw offsets are all set to zero. For the remaining turbines, we pick yaw offsets that incur no negative profits, i.e., for any $k = 2, \ldots, n$, turbine $i \in R_k \setminus (\Delta(n) \cup \{\Delta_n + 1\})$ is assigned the same yaw offset as turbine $i - k + 1$, respectively, all with profit 0. Since this way, every turbine is assigned exactly one yaw offset, $\hat{x}$ indeed describes a feasible yaw configuration and by construction, its objective function value corresponds to $f_\omega^\Sigma(\hat{x}) = n = F$. This shows that the constructed WFYP-DEC instance also has a "yes" answer, which completes the proof. □

We remark that the above construction could easily be adapted so that only non-negative terms can occur in the objective[4], as would be the case in our application when focusing solely on power generation. Note also that, due to the generality of the black-box function f_ω^Σ in WFYP-DEC it is unclear whether one could always find a *rational* certificate of an arbitrary "yes" instance, so containment in the complexity class $\mathcal{NP}$ (and thus, $\mathcal{NP}$-completeness) remains open. However, more importantly, we can slightly modify the proof of Theorem A.3 to obtain the following inapproximability result.

Corollary A.4. *There is no polynomial-time α-approximation algorithm for WFYP, for any $\alpha \leq 1$, unless $\mathcal{P} = \mathcal{NP}$.*

Proof. Revisiting the construction from the proof of Theorem A.3 we modify the function values in Eq. (A1) to 0, δ, δ, δ and $-\varepsilon$ for the five cases, respectively, which then establishes the existence of a Hamiltonian circuit in G if and only if there is a

[4]To that end, we can replace the profit for the third and fifth case in Eq. (A1) by $\varepsilon > 0$ and 0, respectively. Then, the construction instance has a solution of value $n + \varepsilon(n^2 - n - 2)/2$ if and only if G is Hamiltonian.

feasible WFYP solution with value $\frac{1}{2}(n^2+n-2)\delta$. Let $\varepsilon := \frac{1}{2}(n^2+n-4)$ and $\delta := 4(\Delta_n+1)\varepsilon = n^4-3n^2+6n-8$. Then, if the original HCP instance was a "yes"-instance, any non-optimal solution of the constructed WFYP instance has solution value at most $\frac{1}{2}(n^2+n-4)\delta-\varepsilon$.[5] Now suppose there exists a polynomial-time α-approximation algorithm for some arbitrary $1/(n^2+n-1) < \alpha \leq 1$. Since any non-optimal solution (for a WFYP instance constructed from a HCP "yes"-instance) has value at most $\frac{1}{2}(n^2+n-4)\delta-\varepsilon$, but the α-approximation algorithm outputs a solution with value at least $\alpha\cdot(n^2+n-2)\delta/2$, it can only be the case that the solution computed by the algorithm is non-optimal if

$$\alpha\left(\frac{n^2+n-2}{2}\right)\delta \leq \left(\frac{n^2+n-4}{2}\right)\delta-\varepsilon \quad\Leftrightarrow\quad \alpha \leq \frac{n^2+n-4}{n^2+n-2} - \frac{2\varepsilon}{n^2+n-2} = 1-\frac{2}{n^2+n-2} - \left(1-\frac{2}{n^2+n-2}\right) = 0,$$

which contradicts the prerequisite $\alpha > 1/(n^2+n-2) > 0$. Thus, the α-approximation algorithm does, in fact, always yield an (optimal) solution of value $\alpha(n^2+n-2)\delta/2$ if and only if the input HCP instance was a "yes"-instance. It could therefore be used to decide the existence of a Hamiltonian circuit in polynomial time, contradicting $\mathcal{NP}$-hardness of the HCP. To see that this implies that no polynomial-time α-approximation can exist (provided $\mathcal{P} \neq \mathcal{NP}$) for *any* $0 < \alpha \leq 1$, it suffices to observe that $1/(n^2+n-2) \to 0$ as $n \to \infty$. □

Author contributions. CK and SS deliberated the overarching research goals, wrote the corresponding project proposal, were responsible for the funding acquisition, and had the idea to tackle the wind farm yaw problem by discretization. FBü formulated the integer program of the covering approach, wrote the source code, and designed as well as performed the experiments. FBü and AT deliberated the experimental design and result analysis. FBe and AT construed and wrote the $\mathcal{NP}$-hardness and inapproximability proofs. FBe, FBü and AT wrote and revised the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. FBü's research was funded by the Bundesministerium für Bildung und Forschung (BMBF, German Federal Ministry of Education and Research) under grant number 05M18MBA-MOReNet. FBe's research was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through Priority Programme 1962. The authors would like to thank Robert Scholz (University of Heidelberg, Interdisciplinary Center for Scientific Computing), Ole Falkenberg (IAV GmbH; now with DNV Maritime Software GmbH), Bastian Ritter (Industrial Science GmbH; now with Wölfel Engineering GmbH + Co. KG.) and Axel Schild (IAV GmbH) for fruitful discussions about wind turbines.

[5] The same upper bound holds for the optimal value in case of HCP "no"-instances. Moreover, it is easy to see that $f_\omega^\Sigma(x) > 0$ for any non-trivial feasible x, i.e., any x that yields profit δ for at least one index pair (i,j), which is trivially always achievable, so no reasonable algorithm would ever yield $f_\omega^\Sigma(x) \leq 0$.

References

Annoni, J., Gebraad, P. M. O., Scholbrock, A. K., Fleming, P. A., and van Wingerden, J.-W.: Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model, Wind Energy, 19, 1135–1150, https://doi.org/10.1002/we.1891, 2016.

Annoni, J., Fleming, P., Scholbrock, A., Roadman, J., Dana, S., Adcock, C., Porté-Agel, F., Raach, S., Haizmann, F., and Schlipf, D.: Analysis of control-oriented wake modeling tools using lidar field results, Wind Energy Science, 3, 819–831, https://doi.org/10.5194/wes-3-819-2018, 2018.

Annoni, J., Bay, C., Johnson, K., Dall'Anese, E., Quon, E., Kemper, T., and Fleming, P.: Wind direction estimation using SCADA data with consensus-based optimization, Wind Energy Science, 4, 355–368, https://doi.org/10.5194/wes-4-355-2019, 2019.

Bastankhah, M. and Porté-Agel, F.: Experimental and theoretical study of wind turbine wakes in yawed conditions, Journal of Fluid Mechanics, 806, 506–541, https://doi.org/10.1017/jfm.2016.595, 2016.

Bay, C. J., Fleming, P., Doekemeijer, B., King, J., Churchfield, M., and Mudafort, R.: Addressing deep array effects and impacts to wake steering with the cumulative-curl wake model, Wind Energy Science, 8, 401–419, https://doi.org/10.5194/wes-8-401-2023, 2023.

Bernardoni, F., Ciri, U., Rotea, M. A., and Leonardi, S.: Identification of wind turbine clusters for effective real time yaw control optimization, Journal of Renewable and Sustainable Energy, 13, 043 301, https://doi.org/10.1063/5.0036640, 2021.

Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., van Doornmalen, J., Eifler, L., Gaul, O., Gamrath, G., Gleixner, A., Gottwald, L., Graczyk, C., Halbig, K., Hoen, A., Hojny, C., van der Hulst, R., Koch, T., Lübbecke, M., Maher, S. J., Matter, F., Mühmer, E., Müller, B., Pfetsch, M. E., Rehfeldt, D., Schlein, S., Schlösser, F., Serrano, F., Shinano, Y., Sofranac, B., Turner, M., Vigerske, S., Wegscheider, F., Wellner, P., Weninger, D., and Witzig, J.: The SCIP Optimization Suite 8.0, ZIB-Report 21-41, Zuse Institute Berlin, http://nbn-resolving.de/urn:nbn:de:0297-zib-85309, 2021.

Boersma, S., Doekemeijer, B., Vali, M., Meyers, J., and van Wingerden, J.-W.: A control-oriented dynamic wind farm model: WFSim, Wind Energy Science, 3, 75–95, https://doi.org/10.5194/wes-3-75-2018, 2018.

Boersma, S., Doekemeijer, B. M., Siniscalchi-Minna, S., and van Wingerden, J.-W.: A constrained wind farm controller providing secondary frequency regulation: An LES study, Renewable Energy, 134, 639–652, https://doi.org/10.1016/j.renene.2018.11.031, 2019a.

Boersma, S., Keviczky, T., and van Wingerden, J.-W.: Stochastic Model Predictive Control: uncertainty impact on wind farm power tracking, in: 2019 American Control Conference (ACC), pp. 4167–4172, https://doi.org/10.23919/ACC.2019.8814475, 2019b.

Churchfield, M., Lee, S., Moriarty, P. J., Martinez, L. A., Leonardi, S., Vijayakumar, G., and Brasseur, J. G.: A Large-Eddy Simulation of Wind-Plant Aerodynamics, in: 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Nashville, Tennessee, https://doi.org/10.2514/6.2012-537, 2012a.

Churchfield, M. J., Lee, S., Michalakes, J., and Moriarty, P. J.: A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics, Journal of Turbulence, 13, 1–23, https://doi.org/10.1080/14685248.2012.668191, 2012b.

Dar, Z., Kar, K., Sahni, O., and Chow, J. H.: Windfarm Power Optimization Using Yaw Angle Control, IEEE Transactions on Sustainable Energy, 8, 104–116, https://doi.org/10.1109/TSTE.2016.2585883, 2017.

Dong, H. and Zhao, X.: Reinforcement Learning-Based Wind Farm Control: Toward Large Farm Applications via Automatic Grouping and Transfer Learning, IEEE Transactions on Industrial Informatics, 19, 11 833–11 845, https://doi.org/10.1109/TII.2023.3252540, 2023.

Dou, B., Qu, T., Lei, L., and Zeng, P.: Optimization of wind turbine yaw angles in a wind farm using a three-dimensional yawed wake model, Energy, 209, 118 415, https://doi.org/https://doi.org/10.1016/j.energy.2020.118415, 2020.

Ennis, B. L., White, J. R., and Paquette, J. A.: Wind turbine blade load characterization under yaw offset at the SWiFT facility, Journal of Physics: Conference Series, 1037, 052 001, https://doi.org/10.1088/1742-6596/1037/5/052001, 2018.

Fischetti, M.: Mathematical Programming Models and Algorithms for Offshore Wind Park Design, Ph.D. thesis, Technical University of Denmark, Copenhagen, https://orbit.dtu.dk/en/publications/9c92d369-d44d-4476-a4f0-85dbac7f362f, 2017.

Fischetti, M.: On the optimized design of next-generation wind farms, European Journal of Operational Research, 291, 862–870, https://doi.org/10.1016/j.ejor.2020.10.048, 2021.

Fischetti, M. and Pisinger, D.: Mathematical Optimization and Algorithms for Offshore Wind Farm Design: An Overview, Business & Information Systems Engineering, 61, 469–485, https://doi.org/10.1007/s12599-018-0538-0, 2019.

Fleming, P. A., Gebraad, P. M. O., van Wingerden, J.-W., Lee, S., Churchfield, M., Scholbrock, A. K., Michalakes, J., Johnson, K., and Moriarty, P. J.: SOWFA Super-Controller: A High-Fidelity Tool for Evaluating Wind Plant Control Approaches, Tech. rep., National Renewable Energy Laboratory (NREL), https://www.osti.gov/biblio/1068611, proceedings of the EWEA 2013, 2013.

Fleming, P. A., Ning, A., Gebraad, P. M. O., and Dykes, K.: Wind plant system engineering through optimization of layout and yaw control, Wind Energy, 19, 329–344, https://doi.org/10.1002/we.1836, 2016.

Fleming, P. A., Stanley, A. P. J., Bay, C. J., King, J., Simley, E., Doekemeijer, B. M., and Mudafort, R.: Serial-Refine Method for Fast Wake-Steering Yaw Optimization, Journ. of Phys.: Conf. Series, 2265, 032 109, https://doi.org/10.1088/1742-6596/2265/3/032109, 2022.

Garey, M. R. and Johnson, D. S.: Computers and Intractability : A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., 1979.

Gebraad, P. M. O. and van Wingerden, J.-W.: A Control-Oriented Dynamic Model for Wakes in Wind Plants, Journal of Physics: Conference Series, 524, 012 186, https://doi.org/10.1088/1742-6596/524/1/012186, 2014.

Gebraad, P. M. O., Teeuwisse, F. W., van Wingerden, J.-W., Fleming, P. A., Ruben, S. D., Marden, J. R., and Pao, L. Y.: A data-driven model for wind plant power optimization by yaw control, in: Am. Contr. Conf., https://doi.org/10.1109/ACC.2014.6859118, 2014.

Gebraad, P. M. O., Fleming, P. A., and van Wingerden, J.-W.: Comparison of actuation methods for wake control in wind plants, in: 2015 American Control Conference (ACC), pp. 1695–1701, https://doi.org/10.1109/ACC.2015.7170977, 2015.

Gebraad, P. M. O., Thomas, J. J., Ning, A., Fleming, P. A., and Dykes, K.: Maximization of the annual energy production of wind power plants by optimization of layout and yaw-based wake control, Wind Energy, 20, 97–107, https://doi.org/10.1002/we.1993, 2017.

Gurobi Optimization, LLC: Gurobi—Gurobi Optimizer Reference Manual, https://www.gurobi.com, accessed: December 2022, 2022.

Hallac, D., Leskovec, J., and Boyd, S.: Network Lasso: Clustering and Optimization in Large Graphs, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pp. 387–396, Association for Computing Machinery, New York, NY, USA, https://doi.org/10.1145/2783258.2783313, 2015.

Hau, E.: Wind Turbines : Fundamentals, Technologies, Application, Economics, Springer, Berlin, 3rd edn., https://doi.org/10.1007/978-3-642-27151-9, 2013.

Howland, M. F., Lele, S. K., and Dabiri, J. O.: Wind farm power optimization through wake steering, Proceedings of the National Academy of Sciences, 116, 14 495–14 500, https://doi.org/10.1073/pnas.1903680116, 2019.

Howland, M. F., Quesada, J. B., Martínez, J. J. P., Larrañaga, F. P., Yadav, N., Chawla, J. S., Sivaram, V., and Dabiri, J. O.: Collective wind farm operation based on a predictive model increases utility-scale energy production, Nature Energy, 7, 818–827, https://doi.org/10.1002/essoar.10510347.1, 2022.

Jensen, N. O.: A Note on Wind Generator Interaction, Tech. Rep. 2411, Risø National Laboratory, 1983.

Jonkman, J., Butterfield, S., Musial, W., and Scott, G.: Definition of a 5-MW Reference Wind Turbine for Offshore System Development, Tech. rep., National Renewable Energy Laboratory, https://doi.org/10.2172/947422, 2009.

Katic, I., Højstrup, J., and Jensen, N. O.: A Simple Model for Cluster Efficiency, in: EWEC'86. Proceedings. Vol. 1, edited by Palz, W. and Sesto, E., pp. 407–410, Raguzzi, A., 1987.

King, J., Fleming, P., King, R., Martínez-Tossas, L. A., Bay, C. J., Mudafort, R., and Simley, E.: Control-oriented model for secondary effects of wake steering, Wind Energy Science, 6, 701–714, https://doi.org/10.5194/wes-6-701-2021, 2021.

Kost, C., Shammugam, S., Jülch, V., Nguyen, H.-T., and Schlegl, T.: Levelized Cost of Electricity : Renewable Energy Technologies, Tech. rep., Fraunhofer Institute for Solar Energy Systems ISE, Freiburg, https://www.ise.fraunhofer.de/content/dam/ise/en/documents/publications/studies/EN2018_Fraunhofer-ISE_LCOE_Renewable_Energy_Technologies.pdf, 2018.

Meyers, J., Bottasso, C., Dykes, K., Fleming, P., Gebraad, P. M. O., Giebel, G., Göçmen, T., and van Wingerden, J.-W.: Wind farm flow control: prospects and challenges, Wind Energy Science, 7, 2271–2306, https://doi.org/10.5194/wes-7-2271-2022, 2022.

NREL: FLORIS. Version 4.2.2, https://github.com/NREL/floris, accessed: December 2024, 2024.

Quick, J., King, J., King, R. N., Hamlington, P. E., and Dykes, K.: Wake steering optimization under uncertainty, Wind Energy Science, 5, 413–426, https://doi.org/10.5194/wes-5-413-2020, 2020.

Ritter, B., Schild, A., Feldt, M., and Konigorski, U.: The design of nonlinear observers for wind turbine dynamic state and parameter estimation, Journal of Physics: Conference Series, 753, 052 029, https://doi.org/10.1088/1742-6596/753/5/052029, 2016.

Ritter, B., Mora, E., Schlicht, T., Schild, A., and Konigorski, U.: Adaptive Sigma-Point Kalman Filtering for Wind Turbine State and Process Noise Estimation, Journal of Physics: Conference Series, 1037, 032 003, https://doi.org/10.1088/1742-6596/1037/3/032003, 2018.

Schrijver, A.: Theory of Linear and Integer Programming, John Wiley & Sons, Chichester, UK, 1986.

Siniscalchi-Minna, S., Bianchi, F. D., Ocampo-Martinez, C., Domínguez-García, J. L., and De Schutter, B.: A non-centralized predictive control strategy for wind farm active power control: A wake-based partitioning approach, Renewable Energy, 150, 656–669, https://doi.org/10.1016/j.renene.2019.12.139, 2020.

Spudić, V. and Baotić, M.: Fast coordinated model predictive control of large-scale distributed systems with single coupling constraint, in: 2013 European Control Conference (ECC), pp. 2783–2788, https://doi.org/10.23919/ECC.2013.6669391, 2013.

Stanley, A. P., Thomas, J., Ning, A., Annoni, J., Dykes, K., and Fleming, P. A.: Gradient-Based Optimization of Wind Farms with Different Turbine Heights, in: 35th Wind Energy Symposium, Am. Inst. of Aeronaut. and Astronaut., https://doi.org/10.2514/6.2017-1619, 2017.

Stanley, A. P. J., Bay, C., Mudafort, R., and Fleming, P.: Fast yaw optimization for wind plant wake steering using Boolean yaw angles, Wind Energy Science, 7, 741–757, https://doi.org/10.5194/wes-7-741-2022, 2022.

Talavera, M. and Shu, F.: Experimental study of turbulence intensity influence on wind turbine performance and wake recovery in a low-speed wind tunnel, Renewable Energy, 109, 363–371, https://doi.org/10.1016/j.renene.2017.03.034, 2017.

Thomas, J., Tingey, E., and Ning, A.: Comparison of two wake models for use in gradient-based wind farm layout optimization, in: 2015 IEEE Conference on Technologies for Sustainability (SusTech), pp. 203–208, IEEE, https://doi.org/10.1109/sustech.2015.7314347, 2015.

van Wingerden, J.-W., Fleming, P. A., Göçmen, T., Eguinoa, I., Doekemeijer, B. M., Dykes, K., Lawson, M. J., Simley, E., King, J., Astrain, D., Iribas, M., Bottasso, C. L., Meyers, J., Raach, S., Kölle, K., and Giebel, G.: Expert Elicitation on Wind Farm Control, Journal of Physics: Conference Series, 1618, 022 025, https://doi.org/10.1088/1742-6596/1618/2/022025, 2020.

Veers, P. S.: Three-dimensional wind simulation, Tech. rep., Sandia National Laboratories, 1988.

Vladislavleva, E., Friedrich, T., Neumann, F., and Wagner, M.: Predicting the energy output of wind farms based on weather data: Important variables and their correlation, Renewable Energy, 50, 236–243, https://doi.org/10.1016/j.renene.2012.06.036, 2013.

Zhang, P. Y., Romero, D. A., Beck, J. C., and Amon, C. H.: Solving wind farm layout optimization with mixed integer programs and constraint programs, EURO Journal on Computational Optimization, 2, 195–219, https://doi.org/10.1007/s13675-014-0024-5, 2014.